

# **Tvarová optimalizace odtěžovaného svahu povrchového dolu**

## **Shape optimization of open-cast mine**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 27. dubna 2010

.....

Rád bych na tomto místě poděkoval panu Ing. Petrovi Beremlijskému, Ph.D. za jeho velkou pomoc, trpělivost a motivaci při řešení této práce.

## Abstrakt

Tato práce se zabývá tzv. dvouúrovňovým programováním, chceme řešit tzv. min-max problém. Min-max problém v této práci sestává ze dvou kroků, prvním je řešení stavové úlohy a následně budeme provádět optimalizační proces pro nehladkou funkci definovanou výstupními parametry stavové úlohy. Stavovou úlohu budeme řešit efektivně pomocí multigridní metody a optimalizační proces bude řešit bundle trust metoda, která je dostatečně robustní pro minimalizaci nehladké funkce.

**Klíčová slova:** dvouúrovňové programování, min-max problém, Richardsonova metoda, multigridní metoda, bundle trust metoda, nehladká funkce

## Abstract

This thesis deals with bilevel programming, so we will solve min-max problem. In min-max we solve state problem with an effective multigrid method and then we will optimize nonsmooth function defined by output parameters from state problem. We compute optimization process by bundle trust method which is robust enough for minimization nonsmooth functions.

**Keywords:** bilevel programming, min-max problem, Richardson method, multigrid method, bundle trust method, nonsmooth function

## Použité značení

$\Omega$  - vyšetřovaná oblast  
 $E$  - Youngův modul  
 $\nu$  - Poissonova konstanta  
 $\Gamma$  - hranice vyšetřované oblasti  
 $\mathbf{x}$  - prostorová proměnná  
 $u$  - stavová proměnná  
 $\lambda, \mu$  - Laméovy koeficienty  
 $\tau_{ij}$  - tenzor napětí  
 $\mathbf{T}$  - hustota plošných sil  
 $e_{ij}$  - tenzor deformace  
 $F_i$  -  $i$ -tá složka objemové síly  
 $\mathbf{o}$  - nulový vektor  
 $\delta_{ij}$  - Kroneckerovo  $\delta$   
 $\sigma$  - spektrum matice  
 $\partial_\alpha$  - parciální derivace podle  $\alpha$   
 $h$  - diskretizační krok (a chápeme jej, jako diskretizační krok hranice  $\Gamma_4$ )  
 $\Theta$  - cenový funkcionál  
 $\alpha$  - návrhová proměnná  
 $U_{ad}$  - množina přípustných návrhových proměnných  
 $\sigma$  - napětí  
 $\nabla$  - gradient  
 $GEM$  - Gaussova eliminační metoda  
 $MG$  - multigridní metoda  
 $\Delta$  - Laplaceův operátor  
 $conv$  - konvexní obal množiny  
 $[\cdot]$  - vektor  
 $\langle \cdot, \cdot \rangle$  - interval  
 $(\cdot, \cdot)$  - skalární součin  
 $\mathbf{n}_i$  -  $i$ -tá složka jednotkové vnější normály k hranici kompaktní množiny

## Obsah

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Úvod</b>   | <b>5</b>  |
| <b>2</b> | <b>Řešená úloha</b>   | <b>6</b>  |
| 2.1      | Rozbor úlohy . . . . .  | 6         |
| 2.2      | Rovnice . . . . .   | 9         |
| 2.3      | Hraniční podmínky . . . . .   | 11        |
| 2.4      | Formulace úlohy . . . . .   | 12        |
| 2.5      | Nástin metod řešení . . . . .   | 13        |
| <b>3</b> | <b>MKP řešení stavové úlohy</b>   | <b>14</b> |
| 3.1      | Definice okrajové úlohy . . . . .   | 14        |
| 3.2      | Odvození slabého řešení . . . . .   | 14        |
| 3.3      | Diskretizace oblasti . . . . .  | 17        |
| 3.4      | Řešení pomocí MKP . . . . .   | 17        |
| 3.5      | Určení napětí na elementech . . . . .   | 20        |
| <b>4</b> | <b>Řešení pomocí multigridní metody</b>   | <b>21</b> |
| 4.1      | Diskretizace . . . . .  | 21        |
| 4.2      | Richardsonova metoda . . . . .  | 21        |
| 4.3      | Konvergence Richardsonovy metody . . . . .  | 23        |
| 4.4      | Rychlost konvergence . . . . .  | 24        |
| 4.5      | Metoda multigrid . . . . .  | 25        |
| 4.6      | Geometrie oblasti . . . . .   | 27        |
| <b>5</b> | <b>Numerický výpočet stavové úlohy</b>  | <b>37</b> |
| 5.1      | Příklady diskretizací pro multigridní metodu . . . . .  | 37        |
| 5.2      | Příklad numerického řešení stavové úlohy . . . . .  | 37        |
| <b>6</b> | <b>Formulace tvarově optimalizační úlohy</b>  | <b>43</b> |
| 6.1      | Numerické řešení stavového problému . . . . .   | 43        |
| 6.2      | Tvarová optimalizace . . . . .  | 44        |
| <b>7</b> | <b>Metody hledání minima</b>  | <b>46</b> |
| 7.1      | Pojmy z nehladké analýzy . . . . .  | 46        |
| 7.2      | Subgradientní metoda . . . . .  | 47        |
| 7.3      | Bundle metoda . . . . .   | 48        |
| 7.4      | Výpočet cenového funkcionálu a subgradientu cenového funkcionálu . . . . .                            | 49        |
| 7.5      | Bundle trust metoda . . . . .   | 49        |
| <b>8</b> | <b>Experimenty s optimalizační úlohou</b>   | <b>51</b> |
| 8.1      | Příklad 1 . . . . .   | 51        |
| 8.2      | Příklad 1- zkoumání závislosti cenové funkce na změně jednotlivých složek návrhové proměnné . . . . . | 54        |

|           |   |           |
|-----------|---|-----------|
| 8.3       | Příklad 2 . . . . .   | 58        |
| 8.4       | Příklad 2- zkoumání závislosti cenové funkce na změně jednotlivých složek návrhové proměnné . . . . . | 59        |
| 8.5       | Grafické uživatelské rozhraní . . . . .   | 60        |
| <b>9</b>  | <b>Závěr</b>  | <b>65</b> |
| <b>10</b> | <b>Reference</b>  | <b>66</b> |

## Seznam obrázků

|    |  |    |
|----|--|----|
| 1  | Posuzovaná situace svahu . . . . .   | 7  |
| 2  | Zvolený souřadný systém . . . . .  | 7  |
| 3  | Příklad Bézierovy křivky (definované třemi body) reprezentující hranici $s$ . . . . .      | 7  |
| 4  | Deformace ve svahu . . . . .   | 9  |
| 5  | Diskretizace oblasti . . . . .   | 17 |
| 6  | Označený element diskretizace . . . . .  | 19 |
| 7  | Nejhrubší možná diskretizace oblasti . . . . .   | 22 |
| 8  | Jemnější diskretizace oblasti . . . . .  | 22 |
| 9  | V cyklus metody multigrid . . . . .  | 26 |
| 10 | Jemnější diskretizace oblasti generovaná metodou multigrid . . . . .                       | 28 |
| 11 | Vizualizace uložení dat . . . . .  | 30 |
| 12 | Potřebné časy na sestavení matic tuhosti v každé úrovni multigridu . . . . .               | 33 |
| 13 | Procentuální vyjádření potřebného času sestavení matic . . . . .                           | 34 |
| 14 | Procentuální vyjádření potřebného času sestavení matic-úprava . . . . .                    | 35 |
| 15 | Příklad zkoumané oblasti 1 . . . . .   | 38 |
| 16 | Příklad zkoumané oblasti 2 . . . . .   | 38 |
| 17 | Příklad zkoumané oblasti 3 . . . . .   | 39 |
| 18 | Příklad zkoumané oblasti 4 . . . . .   | 39 |
| 19 | Příklad zkoumané oblasti 5 . . . . .   | 40 |
| 20 | Zadání stavové úlohy . . . . .   | 40 |
| 21 | Řešení stavové úlohy včetně složky $x$ napětí . . . . .                                    | 41 |
| 22 | Řešení stavové úlohy včetně složky $y$ napětí . . . . .                                    | 41 |
| 23 | Řešení stavové úlohy včetně složky smyku napětí . . . . .                                  | 42 |
| 24 | Možnosti návrhu Bézierovy křivky . . . . .   | 45 |
| 25 | Před optimalizací (Př. 1). Napětí- složka $x$ ; složka $y$ ; složka smyk; Řešení . . . . . | 54 |
| 26 | Po optimalizaci (Př. 1). Napětí- složka $x$ ; složka $y$ ; složka smyk; Řešení . . . . .   | 55 |
| 27 | Analýza citlivosti pro každou proměnnou z příkladu 1 . . . . .                             | 56 |
| 28 | Před optimalizací (Př. 2). Napětí- složka $x$ ; složka $y$ ; složka smyk; Řešení . . . . . | 60 |
| 29 | Po optimalizaci (Př. 2). Napětí- složka $x$ ; složka $y$ ; složka smyk; Řešení . . . . .   | 61 |
| 30 | Analýza citlivosti pro každou proměnnou z příkladu 2 . . . . .                             | 62 |
| 31 | Fragment svahu- Lom Bílina . . . . .   | 63 |
| 32 | GUI . . . . .  | 64 |



## 1 Úvod

V této práci se budeme zabývat problémem tvarové optimalizace dvourozměrného pružného tělesa. Budeme řešit úlohu ve dvou dimenzích (tj. provedeme řez svahem) protože námi vyšetřovaný svah je z homogenního materiálu a uvažujeme stejný profil v každém místě svahu. Budeme hledat optimální tvar jedné z hranic svahu. Na části hranice jsou předepsány Dirichletovy hraniční podmínky a na části Neumannovy. Jako zatížení uvažujeme objemovou sílu způsobenou zemskou gravitací. Chceme najít takový tvar odtěžované hranice svahu, aby byly maximální napětí vzniklé tímto odtěžením nejmenší možné. Úloha je typu bilevel programming, tedy dvouúrovňové programování a to z toho důvodu, že řešíme stavovou úlohu a nad ní úlohu optimalizace. Stavovou úlohu budeme efektivně řešit pomocí multigridní metody. Pro řešení optimalizační úlohy budeme používat bundle trust metodu. Bundle trust metoda slouží pro minimalizaci nespojitě diferencovatelných funkcí a využijeme ji z důvodu, že naše úloha tvarové optimalizace je nespojitě diferencovatelná. Navíc hledané minimum je obvykle v nehladkém bodě, na čemž by řada klasických minimalizačních algoritmů zhavarovala. V kapitole 2 odvodíme diferenciální rovnici reprezentující posuvy hmoty (naše stavová proměnná) v řezu svahu, které hledáme. Dále odvodíme hraniční podmínky a formulujeme úlohu lineární pružnosti. V kapitole 3 definujeme okrajovou úlohu, odvodíme její slabé řešení a nastíníme způsob diskretizace oblasti. Poté si ukážeme použitý matematický model, s jehož pomocí získáme požadované posuvy a dozvíme se, jak lze dopočítat na základě posuvů napětí v každém místě řezu svahu. V kapitole 4 ukážeme, jak budeme tuto stavovou úlohu řešit multigridní metodou a také aplikaci z teorie grafu, kdy využijeme jiný přístup pro tvorbu hierarchie dat pro metodu multigrid. Potom také ukážeme, jak se dá efektivně sestavit matice tuhosti pro náš stavový problém. Uvedeme zde také, jak se vypořádat s problémem interpolace a restrikce v rámci multigridní metody v konfrontaci s Dirichletovými okrajovými podmínkami. V kapitole 5 ukážeme, jaké důsledky má použití multigridní metody pro diskretizaci, vypočteme stavovou úlohu a vyčíslíme jednotlivé složky napětí ve svahu na každém elementu diskretizace. V kapitole 6 budeme formulovat úlohu tvarové optimalizace. V kapitole 7 si uvedeme pojmy z nehladké analýzy, které jsou důležité pro následující subgradientní metodu a bundle metodu. Ukážeme si, jak vypočteme potřebnou funkční hodnotu cenového funkcionálu a subgradient tohoto funkcionálu. Následně ještě ukážeme námi použitou bundle trust metodu. V kapitole 8 si uvedeme dva příklady tvarové optimalizace. V obou případech popíšeme průběh výpočtu a vykreslíme řez svahu před optimalizací a po optimalizaci. Posléze ještě provedeme analýzu citlivosti nalezeného minima na změnu jednotlivých proměnných cenového funkcionálu. Nakonec si ukážeme grafické uživatelské rozhraní pro prezentaci výsledků této práce.

## 2 Řešená úloha

V této kapitole se seznámíme s řešenou úlohou. Budeme formulovat statickou úlohu pro posouzení stability svahu při odtěžování skrývky - viz Obrázek 1. Budeme tedy odvozovat matematický model pro výpočet posuvů homogenního tělesa pro úlohu elastické lineární pružnosti.

1. Rozměry  $K, L, M$  jsou pevně zadány,
2. Profil svahu  $s$  je zadaná funkce prostorových souřadnic (v navazující optimalizační úloze se bude měnit),
3. Odtěžovaná vrstva je homogenní a zpevněná, je charakterizována hustotou  $\rho$ , Youngovým modulem  $E$  a smykovým modulem  $\nu$ ,
4. Na povrchu nepůsobí žádné zatěžující síly,
5. Svislé rozhraní je dostatečně vzdáleno od hranice  $s$ ,
6. Jako objemovou sílu je potřeba uvažovat pouze gravitaci.

### 2.1 Rozbor úlohy

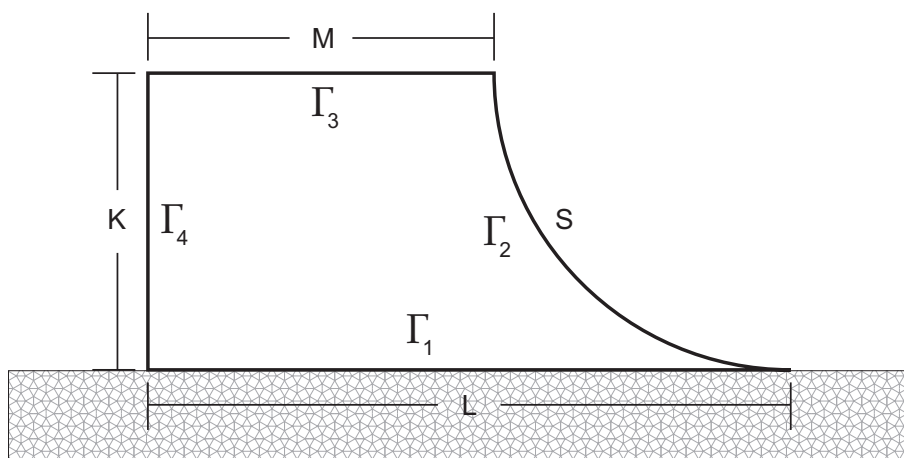
V našem případě se jedná o dvoudimenzionální stacionární problém. Rozměry jsou pevně zadány kromě funkce  $s$  reprezentující hranici svahu, jejíž průběh budeme později optimalizovat. Pro orientaci uvádíme zvolený souřadný systém na Obrázku 2, takže dále budeme psát

$$\mathbf{x} := x_i = (x_1, x_2).$$

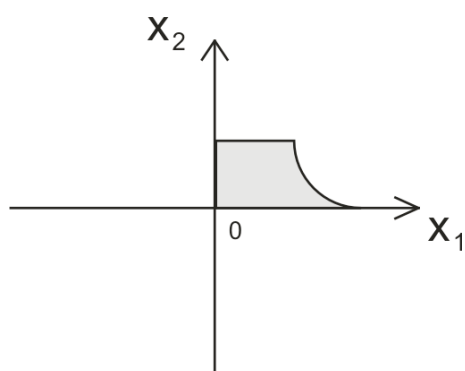
Funkce  $s$  hranice svahu bude v této optimalizační úloze volena jako Bézierova křivka definovaná  $N + 2$  body. Dva krajní body této křivky budou pevně zadány, jeden v patě svahu, což je bod  $[L, 0]$ , druhý bodem je  $[M, K]$ . Zbývající body jsou ve vektoru  $\alpha$ , jehož složky bude možno měnit. Tuto situaci si ukažme na Obrázku 3 pro jednoduchost s jedním řídícím bodem ( $N = 1$ ). Bézierova křivka byla zvolena, protože je hladká a navíc celá tato křivka leží v konvexním obalu bodů, které tuto křivku určují.

Dále je známo, že materiál svahu je homogenní a zpevněný, z toho plyne, že dále budeme uvažovat pružné deformace tuhého tělesa. To znamená, že se těleso nemůže například rozpúlit.

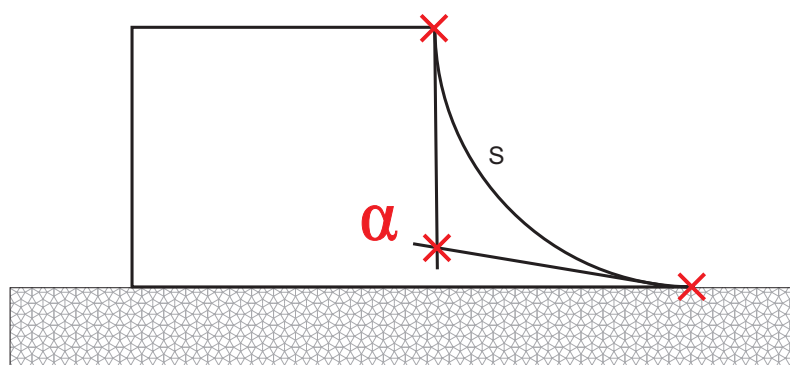
Rozeberme nyní materiálové vlastnosti svahu. Hustota hornin se pohybuje v rozmezí  $\rho = 1,2 - 3,5 \text{ g.cm}^{-3}$ , takže bude vhodné volit hodnoty v tomto intervalu, případně bude si potřeba zvolit konkrétní materiál. U jílu je hustota  $\rho = 1,8 - 2,70 \text{ g.cm}^{-3}$ . Smykový modul, neboli Poissonovo číslo je možno volit kupříkladu pro jíl  $\nu = 0,40 - 0,45$ . Youngův modul pružnosti  $E$  je u zemin ekvivalentní modulu přetvoření  $M$ , jelikož změny vlivem silového působení na materiál jsou nevratné. Jednotkou modulu  $M$  jsou také  $\text{N.m}^{-2} = \text{Pa}$ . Pro tuhý jíl platí



Obrázek 1: Posuzovaná situace svahu



Obrázek 2: Zvolený souřadný systém



Obrázek 3: Příklad Bézierovy křivky (definované třemi body) reprezentující hranici  $s$

$$M = 50 - 150 \text{ kg.cm}^{-2} = (0,5 - 1,5) \cdot 10^5 \text{ N.m}^{-2} = 50 - 150 \text{ kPa}.$$

Definujme pro tento případ tenzor napětí. Ačkoliv uvažujeme kolmé působení sil, ze zadání plyne, že budeme uvažovat síly působící v obou souřadnicích, takže tenzor napětí bude ve tvaru

$$\tau_{ij} = \begin{pmatrix} \tau_{11} & \tau_{12} & 0 \\ \tau_{21} & \tau_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Obecně má tenzor napětí 9 neznámých, avšak ze symetrie je neznámých 6. V našem případě jsou neznámé 3, jelikož jsme ve dvourozměrném prostoru.

Pro popis deformace je podstatný tenzor deformace, jehož tvar je následující

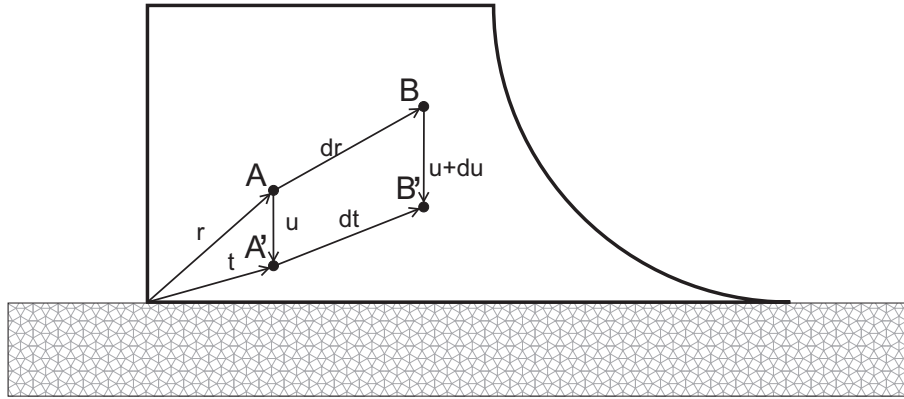
$$e_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

Nyní se budeme zabývat deformací z hlediska geometrie zkoumaného objektu, což vidíme na Obrázku 4, kde :

- $\mathbf{r}, \mathbf{t}$  jsou polohové vektory vzhledem k počátku (zvolen v levém dolním rohu hranice svahu),
- $\mathbf{u}$  je vektor posunutí pro bod  $A$ ,
- $\mathbf{u} + d\mathbf{u}$  je vektor posunutí pro blízký bod  $B$  bodu  $A$ ,
- $\mathbf{t} = \mathbf{r} + \mathbf{u}$ ,
- $\mathbf{t} + d\mathbf{t} = \mathbf{r} + d\mathbf{r} + \mathbf{u} + d\mathbf{u}$ .

Z posledních dvou vztahů vyplývá, že

$$d\mathbf{t} = d\mathbf{r} + d\mathbf{u} \rightarrow dt_i = dr_i + du_i, i = 1, 2.$$



Obrázek 4: Deformace ve svahu

## 2.2 Rovnice

Vektor napětí má následující vlastnosti

- $\mathbf{T}(\mathbf{x}, \mathbf{n}) = -\mathbf{T}(\mathbf{x}, -\mathbf{n})$ ,
- $\mathbf{T}(\mathbf{x}, \mathbf{n}) \mathbf{n} = \mathbf{N}$  (normálové napětí),
- $\mathbf{T}(\mathbf{x}, \mathbf{n}) - \mathbf{N} \mathbf{n} = \mathbf{S}$  (tečné napětí),

kde  $\mathbf{T}$  je hustota plošných sil působících v bodě na hranici oblasti,  $\mathbf{n}$  je normálový vektor na hranici oblasti a  $\mathbf{x}$  je bodem naší oblasti. Bude pro nás podstatné znát souvislost vektoru napětí  $\mathbf{T}(\mathbf{x}, \mathbf{n})$  a tenzoru napětí  $\tau_{ij}(\mathbf{x})$ .

Zavedme si nyní sumační konvenci. Sumační konvence znamená, že, je-li v jednom výrazu zopakovaný index dvakrát, provedeme sumaci přes tyto (shodné) indexy. Např.  $e_{ii} = e_{11} + e_{22}$ , kde  $i = 1, 2$ . S využitím sumační konvence a pro jakékoliv  $\mathbf{x}$  můžeme psát, že

$$\mathbf{T}_i(\mathbf{x}, \mathbf{n}) = \tau_{ij}(\mathbf{x}) \mathbf{n}_j.$$

Pak také pro normálové napětí platí

$$\mathbf{N} = \mathbf{T} \mathbf{n} = T_i \mathbf{n}_j = \tau_{ij} \mathbf{n}_j \mathbf{n}_i.$$

Zapišme rovnováhu sil pro statickou úlohu

$$\frac{\partial \tau_{ij}}{\partial x_j} + F_i = 0. \quad (2.2.1)$$

Připomeňme si tzv. Kroneckerovo  $\delta$ , to je definováno následujícím způsobem

$$\begin{aligned} \delta_{ij} &= 1, \quad i = j, \\ \delta_{ij} &= 0, \quad i \neq j. \end{aligned}$$

Pro napětí (tenzor napětí) platí zobecněný Hookeův zákon pro izotropní těleso

$$\tau_{ij} = \lambda \delta_{ij} \vartheta + 2\mu e_{ij}. \quad (2.2.2)$$

Víme, že pro tenzor deformace platí (zde předpokládáme malé deformace, tedy kvadratický člen je zanedbán a pro větší jednoduchost zápisu přejdeme ze zvýrazněného vektorového značení  $\mathbf{u}$  na označení  $u$ )

$$e_{jk} = \frac{1}{2} \left( \frac{\partial u_j}{\partial x_k} + \frac{\partial u_k}{\partial x_j} \right). \quad (2.2.3)$$

kde  $\lambda$  a  $\mu$  jsou Laméovy koeficienty, jejichž hodnoty se dá vyjádřit pomocí elastických modulů  $E$  a  $\nu$  podle vztahů

$$\mu = \frac{E}{2(1+\sigma)},$$

$$\lambda = \frac{\sigma E}{(1+\sigma)(1-2\sigma)}$$

a

$$\vartheta = e_{11} + e_{22} = e_{ii}.$$

Do (2.2.1) dosadíme (2.2.2), takže

$$\frac{\partial}{\partial x_j} (\lambda \delta_{ij} \vartheta + 2\mu e_{ij}) + F_i = 0.$$

Dále dosadíme (2.2.3), pak

$$\frac{\partial}{\partial x_j} (\lambda \delta_{ij} \vartheta) + \frac{\partial}{\partial x_j} \left[ \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + F_i = 0.$$

Nyní po dosazení za  $\vartheta$  dostáváme obecnou Laméovu rovnici (všimněme si formálního přeznačení indexů u prvního členu)

$$\frac{\partial}{\partial x_i} \left( \lambda \frac{\partial u_j}{\partial x_j} \right) + \frac{\partial}{\partial x_j} \left[ \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + F_i = 0.$$

Máme-li  $\lambda$  a  $\mu$  zadány jako konstanty obecná rovnice bude vypadat takto

$$\lambda \frac{\partial^2 u_j}{\partial x_i \partial x_j} + \mu \frac{\partial^2 u_j}{\partial x_i \partial x_j} + \mu \frac{\partial^2 u_i}{\partial x_j^2} + F_i = 0,$$

$$(\lambda + \mu) \frac{\partial^2 u_j}{\partial x_i \partial x_j} + \mu \frac{\partial^2 u_i}{\partial x_j^2} + F_i = 0,$$

$$(\lambda + \mu) \frac{\partial}{\partial x_i} \left( \frac{\partial u_j}{\partial x_j} \right) + \mu \frac{\partial^2}{\partial x_j^2} (u_i) + F_i = 0,$$

což lze také zapsat vektorově

$$(\lambda + \mu) \operatorname{grad} \operatorname{div} \mathbf{u} + \mu \Delta \mathbf{u} + \mathbf{F} = 0.$$

Výsledkem je tedy lineární parciální diferenciální rovnice druhého řádu, nehomogenní a stacionární.

### 2.3 Hraniční podmínky

Nyní odvodíme hraniční podmínky problému, kde jednotlivé hranice je možno vidět na Obrázku 1. Předpokládejme, že

$$\mathbf{u} = u_i = u_i(x_1, x_2) = (u_1(x_1, x_2), u_2(x_1, x_2)).$$

Odvoďme tedy podmínky pro každou hranici zvlášť takto

- $\Gamma_1$ : budeme přirozeně předpokládat nulové posuvy ve vertikálním směru a nulové působící síly v horizontálním směru, tedy

$$\begin{aligned} u_2(x_1, 0) &= 0, \\ T_1 = 0 &\Rightarrow T_1(\bar{x}, e_j) = \tau_{1j} n_j, \\ \tau_{1j} &= 0, \\ \lambda e_{11} + 2\mu e_{1j} &= 0, \end{aligned}$$

- $\Gamma_2$ : z nulových působících sil na hranici  $\Gamma_2$  předpokládáme nulová napětí, z čehož plynou nulové deformace, takže postupnými úpravami obdržíme

$$\begin{aligned} T_i = 0 &\Rightarrow T_i(\bar{x}, e_j) = \tau_{ij} n_j, \\ \tau_{ij} &= 0, \\ \lambda \delta_{ij} \vartheta + 2\mu e_{ij} &= 0, \end{aligned}$$

- $\Gamma_3$ : bude situace totožná jak v případě  $\Gamma_2$ ,
- $\Gamma_4$ : zde přímo plynou z předpokladů nulové posuvy

$$u_1(0, x_2) = 0; \quad u_2(0, x_2) = 0.$$

Ještě je potřeba provést diskuzi o vstupních datech. Předpokládáme homogenní zpevněný materiál svahu a neuvažujeme pukliny, z toho plyne spojitá objemová síla, takže  $F_i$  je spojitá. Známe také Youngův modul  $E$  a smykový modul  $\nu$ , jsou to konstanty, takže z dříve uvedených vztahů lze pomocí těchto konstant  $\lambda$  a  $\mu$  vypočítat. Derivace stavové proměnné  $u_i$  budou spojité, protože předpokládáme, že těleso zůstane vcelku.

## 2.4 Formulace úlohy

Nyní lze formulovat okrajovou úlohu pro danou situaci svahu.

$$\begin{aligned}(\lambda + \mu) \frac{\partial}{\partial x_1} \left( \frac{\partial u_j}{\partial x_j} \right) + \mu \frac{\partial^2}{\partial x_j^2} (u_1) + F_1 &= 0, \\(\lambda + \mu) \frac{\partial}{\partial x_2} \left( \frac{\partial u_j}{\partial x_j} \right) + \mu \frac{\partial^2}{\partial x_j^2} (u_2) + F_2 &= 0.\end{aligned}$$

s hraničními podmínkami ve tvaru

- $\Gamma_1$ :

$$\begin{aligned}u_2 &= 0, \\ \nabla u_1 &= o.\end{aligned}$$

- $\Gamma_2$ :

$$\begin{aligned}\nabla u_1 &= o, \\ \nabla u_2 &= o.\end{aligned}$$

- $\Gamma_3$ :

$$\begin{aligned}\nabla u_1 &= o, \\ \nabla u_2 &= o.\end{aligned}$$

- $\Gamma_4$ :

$$\begin{aligned}u_1 &= 0, \\ u_2 &= 0.\end{aligned}$$



## 2.5 Nástin metod řešení

Numericky:

Je možno využít MKP, ale další numerickou možností řešení je metoda Single random variable, kde se uvažuje statistická variabilita vstupních dat pro výpočet, přičemž dbáme toho, aby data pocházela z normálního rozdělení. Tato metoda však neuvažuje vztah mezi vlastnostmi elementů zeminy a jejich vzájemnou pozicí v prostoru. Tento problém řeší metoda Random finite element, která z předchozí metody přímo vychází, ale je doplněna o vstup v podobě třetí proměnné charakterizující geologické vlastnosti daného prostředí. My se však budeme zabývat řešením pomocí MKP ve 2D metodou rovinné deformace (v překladu plane strain).

Analyticky:

Pro analytický výpočet lze využít Airyho funkcí. Nástin provedu pro situaci, kde neuvažujeme objemové síly. Zde je situace poměrně komplikovaná, jelikož je potřeba zohlednit Saint-Venantovy rovnice kompatibility, v nich provést substituci a pak jsme schopni vyjádřit napěťové složky přímo pomocí tzv. Airyho funkce. Vyjdeme-li zase ze Saint-Venantových rovnic kompatibility, dosazením Airyho funkcí napětí dostaneme biharmonickou diferenciální rovnici. V tomto případě řešení najdeme bez závislosti na materiálových konstantách, protože ty v dané rovnici nevystupují. Je možno však substitucí za složky napětí dojít k Laplaceově úloze, kde nejprve najdeme řešení pro substituovanou proměnnou a pak vyřešíme soustavu diferenciálních rovnic pro substituce. Druhou možností pro výpočet je využití Greenovy funkce, která ale zohledňuje geometrické vlastnosti hranice tělesa. Už z tohoto předpokladu plyne, že nalezení řešení pro složitější geometrii hranice by bylo nemožné.

### 3 MKP řešení stavové úlohy

#### 3.1 Definice okrajové úlohy

Definujme nejdříve

$$\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4.$$

Definujme oblast  $\Omega$

$$\Omega = \text{int}(\Gamma).$$

Zřejmě pak platí

$$\partial\Omega = (\Gamma).$$

Navíc je zřejmé

$$\overline{\Omega} = \Omega \cup \partial\Omega.$$

Připomeňme si vektorový zápis zadané úlohy

$$(\lambda + \mu) \text{grad div} \mathbf{u} + \mu \Delta \mathbf{u} + \mathbf{F} = 0,$$

kde

$$\mathbf{u} = u = u_i = u_i(x_1, x_2) = (u_1(x_1, x_2), u_2(x_1, x_2)),$$

$$\mathbf{F} = F = F_i = (F_1, F_2) \in L^2(\Omega).$$

Formulujme okrajovou úlohu, jejíž slabé řešení budeme následně odvozovat.

$$\left\{ \begin{array}{ll} (\lambda + \mu) \frac{\partial}{\partial x_i} \left( \frac{\partial u_j}{\partial x_j} \right) + \mu \frac{\partial^2}{\partial x_j^2} (u_i) + F_i & = 0 \quad (x_1, x_2) \in \Omega; \quad i, j = 1, 2 \\ \nabla u_1 & = 0 \quad \forall x_1, x_2 \in \Gamma_1 \\ u_2 & = 0 \quad \forall x_1, x_2 \in \Gamma_1 \\ \nabla u_1 & = 0 \quad \forall x_1, x_2 \in \Gamma_2, \Gamma_3 \\ \nabla u_2 & = 0 \quad \forall x_1, x_2 \in \Gamma_2, \Gamma_3 \\ u & = 0 \quad \forall x_1, x_2 \in \Gamma_4 \end{array} \right. \quad (3.1.1)$$

#### 3.2 Odvození slabého řešení

Definujme nyní prostor testovacích funkcí  $V$  (křivkové integrály chápeme ve smyslu stop)

$$V = \left\{ v(x, y) \in H^1(\Omega) : \int_{\Omega} (v(x, y))^2 + (\nabla v(x, y))^2 dx dy < \infty, \int_{\Gamma_1} v d\Gamma_1 = 0, \int_{\Gamma_4} v d\Gamma_4 = 0 \right\}$$

Mějme na paměti, že  $H^1(\Omega) = W^{1,2}(\Omega)$ .

**Definice 3.1** Buď  $((u, v)) = a(u, v)$ , řekneme, že funkce  $u \in W^{1,2}(\Omega)$  je slabým řešením okrajové úlohy, jestliže

$$u - u_0 \in V, \quad u_0 \in W^{1,2}(\Omega)$$

$$((u, v)) = \int_{\Omega} f(x) v(x) dx + g(v), \quad v \in V, \quad f \in L^2(\Omega),$$

kde  $a(u, v)$  je bilineární forma reprezentující Laplaceův operátor a  $g(v)$  je forma příslušná Neumannovým okrajovým podmínkám.

Připomeňme si zde Greenovu větu, které později využijeme v odvození.

**Věta 3.1** Buď  $u \in W^{1,2}(\Omega)$ ,  $v \in V$  a  $\mathbf{n}_i$  je  $i$ -tá složka vektoru jednotkové vnější normály k hranici oblasti  $\Omega$  a  $\Gamma$  je hranicí oblasti  $\Omega$ , pak platí Greenova věta

$$\int_{\Omega} u(\mathbf{x}) \frac{\partial v(\mathbf{x})}{\partial x_i} d\Omega = - \int_{\Omega} \frac{\partial u(\mathbf{x})}{\partial x_i} v(\mathbf{x}) d\Omega + \int_{\Gamma} u(\mathbf{x}) v(\mathbf{x}) \mathbf{n}_i d\Gamma.$$

Vynásobme nyní diferenciální rovnici z úlohy (3.1.1) funkcí  $v \in V$  a integrujme přes oblast  $\Omega$

$$-(\lambda + \mu) \int_{\Omega} \frac{\partial}{\partial x_i} \left( \frac{\partial u_j}{\partial x_j} \right) v d\Omega - \mu \int_{\Omega} \frac{\partial^2 u_i}{\partial x_j^2} v d\Omega = \int_{\Omega} F_i v d\Omega$$

Následnou aplikací Greenovy věty

$$(\lambda + \mu) \left[ \int_{\Omega} \frac{\partial u_j}{\partial x_j} \frac{\partial v}{\partial x_i} d\Omega - \int_{\Gamma} \frac{\partial u_j}{\partial x_j} v \mathbf{n}_i d\Gamma \right]$$

$$+ \mu \left[ \int_{\Omega} \frac{\partial u_i}{\partial x_j} \frac{\partial v}{\partial x_j} d\Omega - \int_{\Gamma} \frac{\partial u_i}{\partial x_j} v \mathbf{n}_j d\Gamma \right] = \int_{\Omega} F_i d\Omega$$

Rozepíšeme-li rovnici pro čtyři disjunktní části hranice, dostáváme

$$(\lambda + \mu) \left[ \int_{\Omega} \frac{\partial u_j}{\partial x_j} \frac{\partial v}{\partial x_i} d\Omega - \sum_{p=1}^4 \left\{ \int_{\Gamma_p} \frac{\partial u_j}{\partial x_j} v \mathbf{n}_i d\Gamma_p \right\} \right]$$

$$+ \mu \left[ \int_{\Omega} \frac{\partial u_i}{\partial x_j} \frac{\partial v}{\partial x_j} d\Omega - \sum_{p=1}^4 \left\{ \int_{\Gamma_p} \frac{\partial u_i}{\partial x_j} v \mathbf{n}_j d\Gamma_p \right\} \right] = \int_{\Omega} F_i d\Omega$$

Nyní se budeme zabývat dvěma případy, které mohou nastat. Je nutno rozepsat případy pro  $i = 1$  a  $i = 2$  a poté zohlednit okrajové podmínky. Zabývejme se nyní případem, kdy  $i = 1$ .

$$\begin{aligned}
 & (\lambda + \mu) \left[ \int_{\Omega} \left( \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right) \frac{\partial v}{\partial x_1} d\Omega - \sum_{p=1}^4 \left\{ \int_{\Gamma_p} \left( \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right) v \mathbf{n}_1 d\Gamma_p \right\} \right] \\
 & + \mu \left[ \int_{\Omega} \frac{\partial u_1}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\partial u_1}{\partial x_2} \frac{\partial v}{\partial x_2} d\Omega - \sum_{p=1}^4 \left\{ \int_{\Gamma_p} \left( \frac{\partial u_1}{\partial x_1} \mathbf{n}_1 + \frac{\partial u_1}{\partial x_2} \mathbf{n}_2 \right) v d\Gamma_p \right\} \right] = \\
 & = \int_{\Omega} F_1 d\Omega
 \end{aligned}$$

Uplatněním okrajových podmínek na křivkové integrály pro příslušné části hranice dostáváme

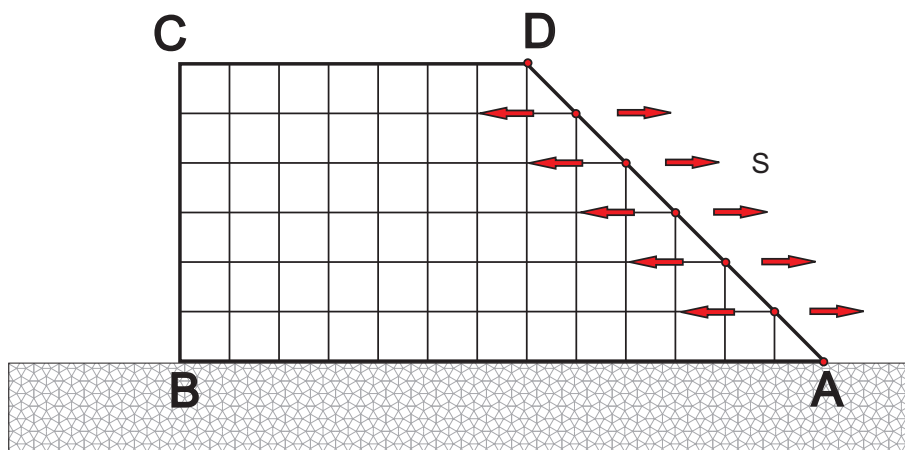
$$(\lambda + \mu) \left[ \int_{\Omega} \left( \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right) \frac{\partial v}{\partial x_1} d\Omega \right] + \mu \left[ \int_{\Omega} \nabla u_1 \nabla v d\Omega \right] = \int_{\Omega} F_1 d\Omega$$

Zbývá nám případ, kdy  $i = 2$

$$\begin{aligned}
 & (\lambda + \mu) \left[ \int_{\Omega} \left( \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right) \frac{\partial v}{\partial x_2} d\Omega - \sum_{p=1}^4 \left\{ \int_{\Gamma_p} \left( \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right) v \mathbf{n}_2 d\Gamma_p \right\} \right] \\
 & + \mu \left[ \int_{\Omega} \frac{\partial u_2}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \frac{\partial v}{\partial x_2} d\Omega - \sum_{p=1}^4 \left\{ \int_{\Gamma_p} \left( \frac{\partial u_2}{\partial x_1} \mathbf{n}_1 + \frac{\partial u_2}{\partial x_2} \mathbf{n}_2 \right) v d\Gamma_p \right\} \right] = \\
 & = \int_{\Omega} F_2 d\Omega
 \end{aligned}$$

Opět zohledníme okrajové podmínky a dostáváme tvar

$$(\lambda + \mu) \left[ \int_{\Omega} \left( \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right) \frac{\partial v}{\partial x_2} d\Omega \right] + \mu \left[ \int_{\Omega} \nabla u_2 \nabla v d\Omega \right] = \int_{\Omega} F_2 d\Omega$$



Obrázek 5: Diskretizace oblasti

### 3.3 Diskretizace oblasti

Pro zadání geometrie objektu byly zvoleny 4 body ( $A, B, C, D$ , kde  $A = [L, 0]$ ,  $B = [0, 0]$ ,  $C = [0, H]$ ,  $D = [D, H]$ ), jak vidíme na schématickém Obrázku 5. Tyto body definují rozměry svahu. Hranici svahu  $s$  budeme modelovat pomocí Bézierovy křivky, připomeňme si tedy předpis, kterým je popsána Bézierova křivka v následujícím Algoritmu 1. Výstupem algoritmu je vektor hodnot  $s\_bound$  odpovídajících hodnotám  $x_1$  souřadnic Bézierovy křivky.

---

#### Algorithm 1 Algoritmus- Bézierova křivka

---

```
function  $[s\_bound] = \text{BezierDeCasteljau}(P, u, n)$ 
  for  $k = 1$  to  $n$  do
     $k = \text{zeros}(\text{length}(P) - 1, 2)$ ;
    for  $i = \text{length}(P)$  to  $-1 : 2$  do
       $P_{i+1,k+1} = P(i+1, :)$ ;
       $P_{i,k+1} = P(i, :)$ ;
       $P_{i,k} = (1 - u) * P_{i+1,k+1} + u * P_{i,k+1}$ ;
       $P_k(i+1, :) = P_{i,k}$ ;
    end for
     $P = P_k$ ;
  end for
   $s\_bound = P$ ;
```

---

### 3.4 Řešení pomocí MKP

Našemu odvozenému slabému řešení odpovídá formulace z knihy The Finite Element Method in Engineering Science od O.C.Zienkiewicz - [2], takže jsme mohli použít vzorce

z knihy, které si zde uvedeme. Budeme se zabývat rovinnou deformací, což najdeme v knize pod názvem rovinné deformace (plane strain). Vyjděme ze základní rovnice pro posunutí (uvádíme i analogii se značením, které budeme používat)

$$\left( \int [B]^T [D] [B] d(vol) \right) \{\delta\}^e = \{F\}^e,$$

$$Ku = f,$$

kde  $F$  označuje objemovou sílu na elementu,  $B$ ,  $D$  jsou matice, jejichž násobením získáme matici tuhosti,  $d(vol)$  označuje integraci přes objem a  $\delta^e$  označuje posuvy na jednotlivých příslušných uzlech v tomto smyslu

$$\delta^e = \begin{Bmatrix} \delta_i \\ \delta_j \\ \delta_m \end{Bmatrix} = \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix}$$

Budeme potřebovat získat matice pro sestavení globální matice tuhosti a vektor objemových sil, zde si uveďme odvozené vztahy z [2]. Víme, že jsme v elementu  $e$  daného vrcholy v pořadí proti směru hodinových ručiček s indexy  $i, j, m$ , tedy máme trojúhelník s uzly příslušnými elementu, což vidíme na Obrázku 6.

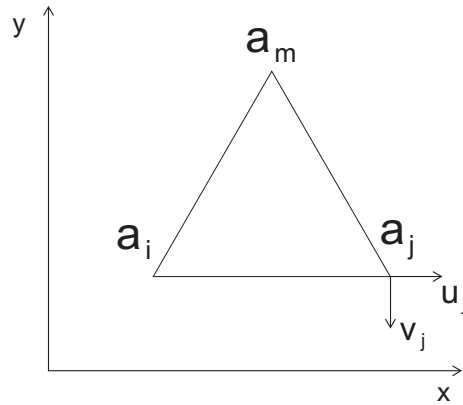
$$a^e = \begin{Bmatrix} a_i \\ a_j \\ a_m \end{Bmatrix} = \begin{Bmatrix} x_i \\ y_i \\ x_j \\ y_j \\ x_m \\ y_m \end{Bmatrix}$$

Nyní si označme nové koeficienty  $a, b, c$ , pomocí nichž jsou pak odvozovány matice  $B$  a  $D$ .

$$a_i = x_j y_m - x_m y_j$$

$$b_i = y_j - y_m$$

$$c_i = x_m - x_j$$



Obrázek 6: Označený element diskretizace

a ostatními koeficienty získanými cyklickou permutací indexů pořadí i,j,m.

$$2\Delta = \det \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix}.$$

Připomeňme zde, že  $E$  je Youngův modul pružnosti a  $\nu$  je smykový modul

$$B = \frac{1}{2\Delta} \begin{bmatrix} b_i & 0 & b_j & 0 & b_m & 0 \\ 0 & c_i & 0 & c_j & 0 & c_m \\ c_i & b_i & c_j & b_j & c_m & b_m \end{bmatrix}.$$

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix}.$$

Potom můžeme definovat lokální matici tuhosti na elementu  $e$  takto

$$k = \begin{bmatrix} k_{ii} & k_{ij} & k_{im} \\ k_{ji} & k_{jj} & k_{jm} \\ k_{mi} & k_{mj} & k_{mm} \end{bmatrix},$$

kde

$$[k_{rs}] = [B_r]^T [D] [B_s] t\Delta$$

je submatice s indexy  $r, s = 1, 2, 3$ . To znamená, že každá submatice má čtyři prvky, submatice máme devět, tedy lokální matice tuhosti sestává z 36 prvků:  $k \in \mathbb{R}^{6 \times 6}$ . Globální

matici tuhosti dostaneme dosazením všech devíti submatic lokální matice tuhosti na pozice příslušné daným uzlům.

Nyní se zabývejme objemovou silou. V naší úloze neuvažujeme žádné vnější vlivy ani počáteční deformace či napětí, takže budeme brát v potaz distribuované objemové síly, které vyjádříme následující rovnicí

$$\{F\}_p^e = - \begin{pmatrix} X_i \\ Y_i \\ X_j \\ Y_j \\ X_m \\ Y_m \end{pmatrix} \frac{\Delta}{3},$$

kde se objevuje 6 hodnot a každá je příslušná jednomu elementu tak, jak jsme uvedli dříve v pořadí proti hodinovým ručičkám jsou hodnoty po dvou pro každý uzel a vyjadřují objemovou sílu v dané souřadnici.

### 3.5 Určení napětí na elementech

Pro naši optimalizační úlohu budeme potřebovat vyjádřit mechanické napětí na elementech. To je možno dopočítat po vyřešení stavové úlohy, jejímž výstupem budou posuvy pro každý uzel ve zvolené diskretizaci. Vyjádřit napětí je možno pomocí výše definovaných matic  $B$ ,  $D$  a posuvů. Dostaneme výstup ve formě vektoru se třemi prvky. Jejich význam definuje následující rovnice

$$\{\sigma\} = \begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix},$$

což odpovídá napětí ve směru osy  $x, y$  a smykovému napětí  $\tau_{xy}$ . Vyjádření těchto napětí je možné pomocí rovnice

$$\{\sigma\} = [D][B]\{\delta\}^e. \quad (3.5.1)$$

To nám dává možnost po vypočtení stavové úlohy zjistit napětí na vybraných elementech- tento výpočet je možné provádět lokálně pouze tam, kde nás napětí zajímají. Tato skutečnost bude podstatná pro naši optimalizační úlohu, kterou budeme zkoumat v pozdějších kapitolách.



## 4 Řešení pomocí multigradní metody

### 4.1 Diskretizace

Podobu diskretizované oblasti jsme do této chvíle měli uvedenou jen pro představu. Je však důležité takovou diskretizaci utvořit s ohledem na potřeby numerické metody. Na Obrázku 7 vidíme, jak vypadá konkrétní triangulace na nejhrubší síti, kterou budeme používat. Všimněme si také označení uzlů- ty jsou popsány čísly modré barvy a samotné elementy jsou označeny číslem červené barvy. Můžeme si také ukázat jemnější síť. I hranice svahu, kterou jsme si dříve nazvali  $s$  tímto bude také hladší. Je to dáno tím, že hranice je propočítávána adaptivně pomocí parametrů zadaných metodě, která pomocí Bézierovy křivky napočítá body na hranici. Po dosažení požadovaných bodů však získáme po částech lineární křivku. Na Obrázku 8 tak vidíme, jak vypadá oblast vygenerovaná s jemnějším krokem.

Takto lze diskretizovat oblast pro numerickou metodu s takovým jemným diskretizačním krokem  $h$ , jaký budeme požadovat. Diskretizačním krokem rozumíme diskretizaci na hranici  $\Gamma_1$ , pak diskretizace na celém zbytku oblasti je přizpůsobena rozměrům oblasti. Na základě takové triangulace si připravíme matici tuhosti  $K$  a pak vektor pravé strany  $f$ . Taková soustava by se dala řešit pomocí přímé metody, jako je například Gaussova eliminační metoda. Ta je však pro naše potřeby příliš pomalá, patří totiž do třídy složitosti  $O(n^3)$ . Proto budeme náš problém řešit pomocí iterační metody, která nám poskytne řešení se zadanou přesností, ale s nižšími časovými i paměťovými nároky.

### 4.2 Richardsonova metoda

Budeme se nyní zabývat Richardsonovou iterační metodou. Richardsonova iterační metoda řeší soustavu

$$Ku = f,$$

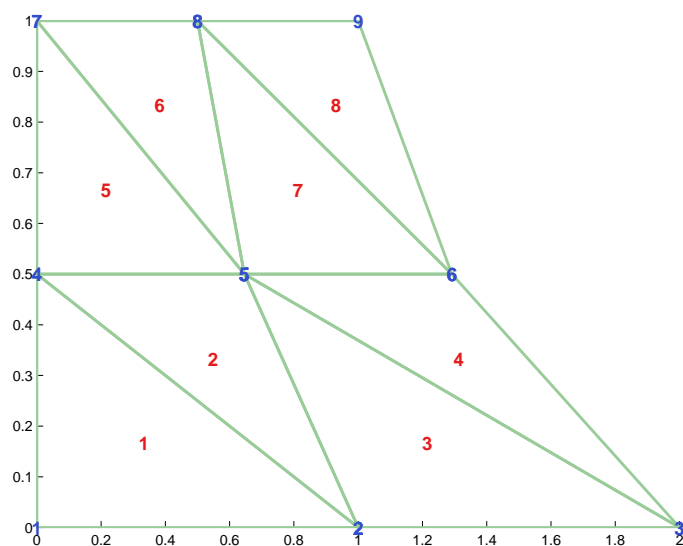
kde vektor  $u$  představuje hledané řešení.

Pro vlastní vektory  $y_i$  tvořící ortonormální bázi  $\mathbb{R}^n$  a vlastní čísla  $\lambda_i$ , kde  $i = (1, 2, \dots, n)$ ,  $n \in \mathbb{N}$  platí

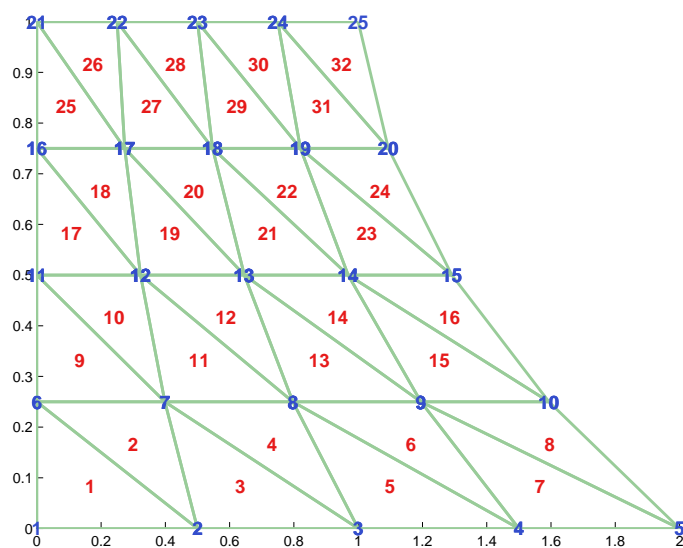
$$\begin{aligned} Ky_i &= \lambda_i y_i, \\ \|y_i\| &= 1, \\ 0 &< \lambda_1 < \lambda_2 < \dots < \lambda_n = \lambda_{max}. \end{aligned}$$

Pro jednoznačnost řešení je nutné, aby matice  $K$  byla symetrická pozitivně definitní matice řádu  $\mathbb{R}^{n \times n}$ ,  $n \in \mathbb{N}$ . Předpis této iterační metody je ve tvaru

$$u^{(k+1)} = u^{(k)} + \omega \left( f - Ku^{(k)} \right), \omega > 0.$$



Obrázek 7: Nejhrubší možná diskretizace oblasti



Obrázek 8: Jemnější diskretizace oblasti

### 4.3 Konvergence Richardsonovy metody

Označme si chybu v každé  $k$ -té iteraci  $e^{(k)}$ , pro ní pak platí vztah

$$e^{(k)} := f - Ku^{(k)}.$$

Takže pak lze psát

$$\begin{aligned} e^{(k+1)} &= f - Ku^{(k+1)} = f - K \left[ u^{(k)} + \omega (f - Ku^{(k)}) \right] = \\ &= e^{(k)} - \omega Ke^{(k)} = e^{(k)} (I - \omega K). \end{aligned}$$

Porovnejme nyní podíl chyby v  $k$ . a  $k + 1$ . iteraci.

$$\frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} \leq \|I - \omega K\| = \max_{\text{spektrum } I - \omega K} |\dot{\sigma}(I - \omega K)| = \max_i |1 - \omega \lambda_i|$$

a aby docházelo ke konvergenci iterační metody, požadujeme

$$\max_i |1 - \omega \lambda_i| < 1$$

Chceme znát horní odhad čísla  $\omega$ , proto si označme  $\omega_{max}$  jako největší koeficient. Pro  $\omega_{max}$  platí

$$|1 - \omega_{max} \lambda_{max}| < 1 \Rightarrow -(1 - \omega_{max} \lambda_{max}) < 1$$

Takže platí

$$\omega_{max} < \frac{2}{\lambda_{max}}.$$

Proto volíme

$$\omega = \frac{2}{\Lambda}, \Lambda > \lambda_{max}.$$

Pak  $\Lambda$  lze volit například z Geršgorinovy věty takto (pro řídké matice má tento výpočet složitost  $O(n)$ )

$$\Lambda = \left\{ a_{ii} + \sum_{j=1; j \neq i}^n |a_{ij}|; i \in \{1, 2, \dots, n\} \right\}.$$

Pro odhad vlastních čísel můžeme také použít knihovní funkci *eig* z Matlabu. Pokusme se najít nejlepší volbu  $\omega$ , označme ji  $\omega_{opt}$  a platí pro ní

$$\begin{aligned} 1 - \omega_{opt} \lambda_{min} &= -(1 - \omega_{opt} \lambda_{max}) \\ 2 &= (\lambda_{min} + \lambda_{max}) \omega_{opt} \\ \omega_{opt} &= \frac{2}{\lambda_{min} + \lambda_{max}}. \end{aligned}$$

#### 4.4 Rychlost konvergence

$$\begin{aligned} \max_i |1 - \omega_{opt} \lambda_i| &= 1 - \omega_{opt} \lambda_{min} = 1 - \frac{2\lambda_{min}}{\lambda_{min} + \lambda_{max}} = \\ &= \frac{\lambda_{min} + \lambda_{max} - 2\lambda_{min}}{\lambda_{min} + \lambda_{max}} = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{min} + \lambda_{max}} \frac{1}{\frac{\lambda_{min}}{\lambda_{max}}} = \\ &= \frac{\frac{\lambda_{max}}{\lambda_{min}} - 1}{\frac{\lambda_{max}}{\lambda_{min}} + 1} = \frac{\kappa(K) - 1}{\kappa(K) + 1}, \end{aligned}$$

kde  $\kappa(K)$  je číslo podmíněnosti matice  $K$ . Tedy pro velké soustavy bude metoda konvergovat velmi pomalu, jelikož právě tyto mají velká čísla podmíněnosti.

Buď

$$\begin{aligned} e^k &= \sum_{i=1}^n \xi_i^k y_i, \\ e^{k+1} &= \sum_{i=1}^n \xi_i^{k+1} y_i. \end{aligned}$$

Dle předpokladů o uspořádání vlastních čísel musí platit

$$\frac{|\xi_i^{k+1}|}{|\xi_i^k|} < 1.$$

Nyní si rozepíšme chybu která vznikne v  $k+1$ . iteraci a vynásobme  $j$ . vlastním vektorem

$$\begin{aligned} e^{k+1} &= e^k (I - \omega K), \\ (e^{k+1})^T y_j &= (e^k)^T (I - \omega K)^T y_j. \end{aligned}$$

Po rozepsání chyby jako lineárních kombinací bázevých vektorů  $y_i, i = \{1, 2, \dots, n\}, n \in \mathbb{N}$  a využití vlastnosti skalárního součinu vektorů, které patří do totožného ortonormálního systému dostaneme

$$\begin{aligned} \underbrace{\sum_{i=1}^n \xi_i^{k+1} y_i^T y_j}_{\xi_j^{k+1}} &= \sum_{i=1}^n \xi_i^k y_i^T \underbrace{(I - \omega K)^T y_j}_{y_j - \omega K y_j = (1 - \omega \lambda_j) y_j} \\ \xi_j^{k+1} &= \sum_i \xi_i^k y_i^T (1 - \omega \lambda_j) y_j \\ \xi_j^{k+1} &= \xi_j^k (1 - \omega \lambda_j), \end{aligned}$$

proto

$$\frac{|\xi_j^{k+1}|}{|\xi_j^k|} = |1 - \omega \lambda_j|.$$

Z předešlého vztahu je patrné, že je-li  $j = n$ , je konvergence nejlepší a naopak, je-li  $j = 1$ , je konvergence nejhorší. Složky odpovídající velkým indexům vlastních čísel reprezentují vysoké frekvence rezidua. Z tohoto plyne, že Richardsonova metoda efektivně eliminuje vysoké frekvence, kdežto u nižších frekvencí, odpovídající nízkým vlastním číslům, dochází k velmi pomalé konvergenci rezidua.

## 4.5 Metoda multigrid

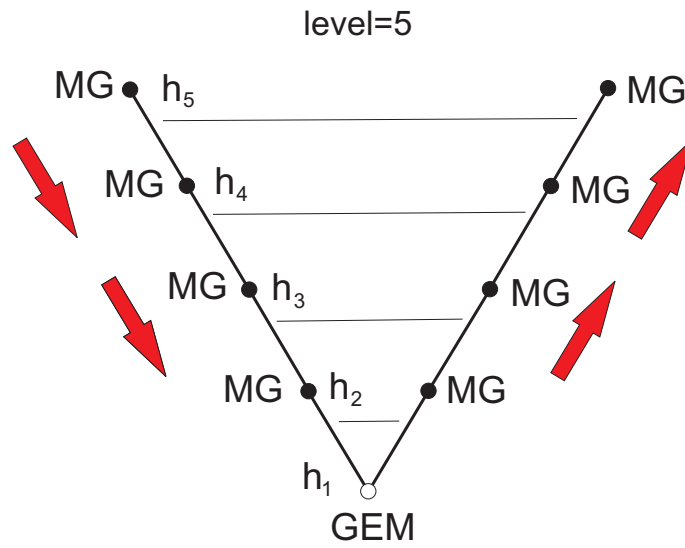
Nápad této metody tkví v tom, že jestli Richardsonova metoda tlumí dobře pouze vyšší frekvence rezidua, potom by bylo výhodné tuto metodu v rámci jednoho problému aplikovat rekurzivně na celé hierarchii diskretizačních kroků. Jestliže Richardsonova metoda rychle odstraní z rezidua oscilace úměrné diskretizaci  $h$ , pak pro nás bude výhodné mít těchto sítí velký počet a tím zajistíme rychlou konvergenci širokého frekvenčního spektra rezidua. Je pouze potřeba uvést způsob, jakým lze přecházet mezi sítěmi. Bude se jednat o tzv. interpolaci a prolongaci vektoru rezidua a také matice tuhosti. K interpolaci dochází při přechodu z jemnější sítě na hrubší a naopak, při přechodu z hrubší na jemnější síť, mluvíme o prolongaci. Na Obrázku 9 vidíme ve zjednodušené podobě, jak probíhá výpočet metody multigrid na pěti úrovních. Takový výpočet se dá schématicky zapsat do tvaru písmene V. Výpočet začíná na nejjemnější síti několika iteracemi ( $m$  iterací) Richardsonovy metody a poté je výsledek interpolován na hrubší diskretizaci a opět je provedeno  $m$  iterací Richardsonovy metody, až se takto dostaneme na hrubou síť, tam je již levné provést výpočet přímou metodou a opět se vracíme zpět na jemnější úroveň, kde výsledek prolongujeme a znova provedeme  $m$  iterací Richardsonovy metody na každé úrovni. Jestliže sestupujeme v tomto výpočtu v tzv. V-cyklu z jemnější na hrubší síť, mluvíme o tzv. presmoothingu- předhlazování a naopak, postupujeme-li ve výpočtu z hrubší na jemnější síť, je řeč o postsmoothingu- opětovné vyhlazování.

Ukažme si algoritmus multigridu na Algoritmu 2.

Jak přecházet mezi úrovněmi diskretizací? Prolongační matice  $I$  slouží k přechodu z diskretizace  $h_i$  na  $h_{i+1}$ , přičemž platí  $h_i = \frac{1}{n_i-1}$ ;  $h_{i+1} = \frac{1}{n_{i+1}-1}$ ,  $h_i = 2h_{i+1}$ ,  $i \in \mathbb{N}$ , pak má prolongační matice dimenzi  $I \in R^{n_{i+1} \times n_i}$ . Buď pro diskretizace  $h_i$  a  $h_{i+1}$  matice tuhosti  $K^{n_i}, K^{n_{i+1}}$  a rezidua pro každou úroveň  $r^{n_i}, r^{n_{i+1}}$ . Pak platí ( $\approx$  značí spektrálně ekvivalentní)

$$K^{n_i} \approx I^T K^{n_{i+1}} I,$$

$$r^{n_i} = I^T r^{n_{i+1}}.$$



Obrázek 9: V cyklus metody multigrid

---

**Algorithm 2** Algoritmus Multigridu

---

```

Multigrid ( $l, r^0$ )
 $r^l = r^0$ 
if  $l=1$  then
     $u^l = K^{l-1} r^l$ 
else
    for  $i = 1$  to  $m$  do
         $u^l = u^l + \omega^l r^l$ 
         $r^l = r^l - \omega K r$ 
    end for
     $r^{l-1} := (I^l)^T r^l$ 
     $v^l = \text{Multigrid}(l-1, r^{l-1})$ 
     $u^l := u^l + I^l v^{l-1}$ 
     $r^l := r^l - K^l I^l v^{l-1}$ 
    for  $i = 1$  to  $m$  do
         $u^l = u^l + \omega^l r^l$ 
         $r^l := r^l - \omega K r$ 
    end for
end if

```

---

Je potom rozumné vytvořit si celou informaci o datech potřebných pro výpočet multigridu, struktury matic a vektorů, a tyto pak uchovávat v globálních polích a posléze spustit samotný výpočet multigridní metody, která již sama bude tyto struktury používat. Klasická příprava těchto struktur pro úroveň  $l$  multigridní metody má tvar

---

**Algorithm 3** Tvorba hierarchie struktur dat pro metodu multigrid
 

---

```

vytvoř první síť  $\Rightarrow$  uzly, hrany, prvky, interpolace
sestav matice tuhosti, pravé strany, odhadni omega  $\Rightarrow K, f, \omega$ 
for  $i = 2$  to  $l$  do
    vytvoř druhou síť pomocí předchozí  $\Rightarrow$  uzly, hrany, prvky, interpolace
    sestav matice tuhosti, pravé strany, odhadni omega  $\Rightarrow K, f, \omega$ 
end for
  
```

---

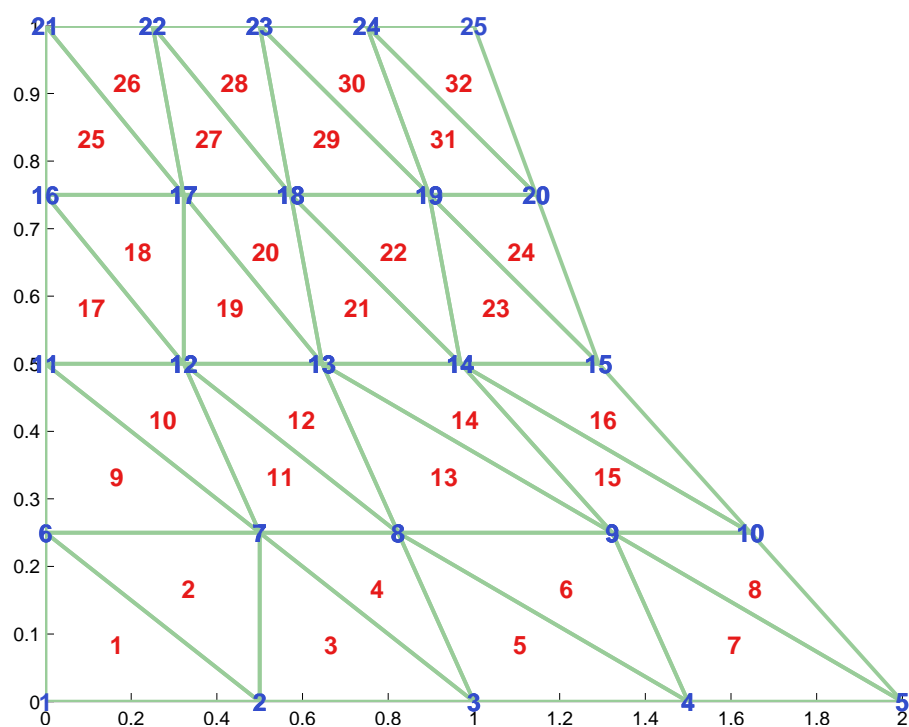
V tomto algoritmu nedocházelo ke třídění. Neměli jsme informaci o tom, který řádek interpolační matice by mohl odpovídat kterému uzlu v diskretizaci. V implementaci se využilo jisté vlastnosti geometrie oblasti, kterou jsme diskretizovali, v důsledku čehož bylo možno dopředu znát strukturu každé další sítě a bylo možno přímo třídit všechna data tak, aby byla umístěna v polích logicky po sobě jdoucích jako v původní diskretizaci.

## 4.6 Geometrie oblasti

Budeme tedy potřebovat ještě další síť kromě jedné původní, takové, které budou mít každou hranu rozdělenou napůl, z níž následně vznikne nový uzel a začlení se do existující triangulace. Tyto nové sítě bude využívat algoritmus multigridu pro svůj hierarchický výpočet na více sítích. Nyní si vystačíme s informací, že potřebujeme novou síť. Nová požadovaná síť pro tu, kterou jsme si již uvedli na Obrázku 7, bude vypadat tak, jak vidíme na Obrázku 10. Je možno všimnout si odlišnosti vůči síti, kterou jsme si dříve uvedli na Obrázku 8. Tato odlišnost je dána tím, že vždy najdeme nové uzly v polovině hran. Proto nedosahujeme takové jemnosti na hranici  $s$ . Je možno si všimnout jiné struktury triangulace uvnitř oblasti, což je důsledkem hledání nových uzlů v polovině hran mezi uzly předchozí triangulace (té, která je o úroveň níže, tedy s hrubší triangulací). Chceme-li tedy větší detaily, musí být iniciální síť pro metodu multigrid jemnější, avšak je vhodné myslet na to, že právě na této nejhrubší síti proběhne GEM výpočet, takže by neměla být až přespříliš jemně volená.

Klasický algoritmus multigridu jsme si již uvedli. V tomto případě se však dalo všimnout určité vlastnosti, která povede k deterministickému výpočtu nové sítě včetně třídění. Budeme se na naši diskretizovanou oblast dívat jako na graf. Buď tedy naše diskretizovaná oblast graf  $G$  a jeho uzly nechť jsou vrcholy  $v_i, i = 1 \dots n, n \in \mathbb{N}$ .

Dá se všimnout, že v grafu se vyskytují vždy vrcholy stejných stupňů. Stupně  $v_i$  (označme  $\deg(v_i)$ ) mohou být 2, 3, 4, 6 a žádné jiné se v tomto grafu neobjeví. Zapišme si pro různé diskretizační kroky (jako diskretizační krok budeme chápat vzdálenost vrcholů nad sebou na hranici  $\Gamma_4$ , tedy délku svislé hrany, délky ostatních hran jsou adaptivně dopočítávány). Tyto počty vrcholů pro každý jednotlivý stupeň vrcholu  $v_i$  si uvedeme následující Tabulkou 1. Protože víme, že v každém řádku bude stejný počet vrcholů,



Obrázek 10: Jemnější diskretizace oblasti generovaná metodou multigrid

| diskretizační krok | stupeň 2 | stupeň 3 | stupeň 4 | stupeň 6 |
|--------------------|----------|----------|----------|----------|
| 0.5                | 2        | 2        | 4        | 1        |
| 0.25               | 2        | 2        | 12       | 9        |
| 0.125              | 2        | 2        | 28       | 49       |

Tabulka 1: Počet uzlů v grafu v závislosti na stupni vrcholu



| diskretizační krok | stupeň 2 | stupeň 3 | stupeň 4 | stupeň 6  |
|--------------------|----------|----------|----------|-----------|
| $h$                | 2        | 2        | $4(n-2)$ | $(n-2)^2$ |

Tabulka 2: Počet uzlů v grafu v závislosti na stupni vrcholu- obecné vztahy

| diskretizační krok | H  | V  | S  |
|--------------------|----|----|----|
| 0.5                | 6  | 6  | 4  |
| 0.25               | 20 | 20 | 16 |
| 0.125              | 72 | 72 | 64 |

Tabulka 3: Množství hran v grafu v závislosti na  $h$ 

taktéž i ve sloupci, řekneme tedy, že vrcholů bude  $n^2$ . Takže pro diskretizační krok  $h$  bude platit  $h = \frac{1}{n-1}$ . Potom lze obecně vyjádřit počet vrcholů každého stupně jako v Tabulce 2. Označme si  $hr(G)$  jako počet hran v grafu  $G$ . Základní věta z teorie grafu nám říká, že pro každý graf  $G$  platí

$$2hr(G) = \sum_{i=1}^n \deg(v_i).$$

Použijme tedy tuto větu a dosaďme naše hodnoty

$$2hr(G) = 2.2 + 2.3 + 4.4.(n-2) + 6.(n-2)^2,$$

dostaneme pak po úpravách

$$hr(G) = 3n^2 - 4n + 1.$$

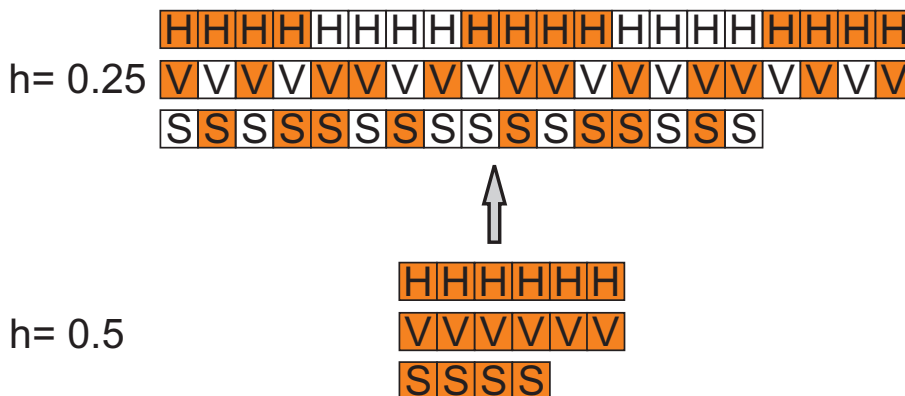
Takže číslo  $hr(G)$  nám označuje, kolik je hran v našem grafu  $G$ . Dalším pohledem na nakreslení grafu  $G$  lze zjistit, že existuje také vztah mezi počtem hran, které si speciálně označíme jako horizontální- podstava trojúhelníku v triangulaci (ozn. H), vertikální (ozn. V) a šikmé- přepona trojúhelníku triangulace (ozn. S). Toto názvosloví je jasné přímo z pohledu na diskretizovanou oblast představující graf  $G$ . Tyto počty jednotlivých hran nám poslouží pro třídění hran v polích, pro naši jednodušší orientaci. V poli se bude nejdříve udržovat pole H hran, pak V a nakonec budou hrany S. Abychom tak ale mohli učinit, potřebujeme vědět, kolik kterých hran bude pro určitý diskretizační krok. Tyto počty máme v Tabulce 3 uvedeny.

Opět lze dojít k obecnějším vztahům v Tabulce 4. Pro kontrolu výsledek sečteme a porovnáme s hodnotou  $hr(G)$

$$n^2 - n + n^2 - n + n^2 - 2n + 1 = 3n^2 - 4n + 1,$$

| diskretizační krok | H         | V         | S              |
|--------------------|-----------|-----------|----------------|
| h                  | $n^2 - n$ | $n^2 - n$ | $n^2 - 2n + 1$ |

Tabulka 4: Množství hran v grafu



Obrázek 11: Vizualizace uložení dat

což je přesně hodnota  $hr(G)$ . Takže máme informaci o tom, kolik bude náš graf  $G$  obsahovat hran H, V a S. To je velmi podstatné pro alokaci polí pro jemnější diskretizace, jinými slovy, jsme schopni dopředu určit, jak velká pole budeme potřebovat pro konstrukce dalších sítí pro potřeby metody multigrid (je totiž obvykle drahé přéalokovávat pole, případně data zbytečně kopírovat). Při plnění těchto polí je však třeba dbát na to, abychom ukládali data tak, aby byla stále setříděná a tím pádem si nechávali prostor i pro nově vzniknuvší uzly, hrany a elementy, případně pole elementů určeného pomocí hran. Graficky si takové uspořádání lze představit například pomocí Obrázku 11. Oranžově jsou vyznačeny objekty, které zůstanou totožné (tedy v tom smyslu, že se pouze rozdělí na dvě části a přemístí do jiných pozic v poli příslušných umístění v grafu) a hrany s bílým pozadím vzniknou nově. Tento případ je reálnou ilustrací konkrétního třídění, přesný zápis by byl velmi nepřehledný, proto zde máme jen schématický náznak. Možností, jak ukládat takové struktury dat je více, záleží na počátečním rozvržení, které je ideální zachovat stále stejné. Je také zřetelné, že nejprve jsou uloženy setříděné hrany H, pak V a nakonec jsou v poli hran hrany S, takže vždy je možno přesně vypočítat, kde najdeme vybranou hranu z grafu.

#### 4.6.1 Efektivní sestavení matice tuhosti

Další podstatnou otázkou bylo, jak sestavit matici tuhosti efektivně. Důvodem byla doba vyskládání matice tuhosti, ta byla příliš vysoká a byl nalezen prostor pro zefektivnění tohoto procesu. Bylo využito faktu, že v Matlabu se uložení matice realizuje vektorově. To znamená, že matice, je v paměti zapsána jako vektor. Kupříkladu, budeme-li mít čtvercovou matici řádu 3, bude v paměti uložena jako vektor délky 9. V našem případě

pracujeme s maticí tuhosti  $K \in \mathbb{R}^{2*n \times 2*n}$ , kde  $n$  je počet uzlů diskretizace. Rozměr je dán reprezentací dat ve vektorech, kdy pro tento 2D problém je na pozici  $i$  x složka a na pozici  $i + 1$  y složka. Proto bude lokální matice tuhosti  $k \in \mathbb{R}^{6 \times 6}$ .

---

**Algorithm 4** Matice tuhosti - klasický algoritmus
 

---

```

n počet neznámých
vytvoř řídkou matici  $K \in \mathbb{R}^{2*n \times 2*n}$ 
for  $i = 1$  to  $n$  do
    vytvoř  $k \in \mathbb{R}^{6 \times 6}$ 
    přiřaď  $K + k$  (znaménko plus je zde pouze symbolické přiřazení příslušných dat z
    lokální matice tuhosti do globální matice tuhosti)
end for
  
```

---

Je evidentní, že se provádí velmi mnoho přístupů do globální matice tuhosti, konkrétně 9 přístupů (to protože máme lokální matici s 36 prvky, a každé 4 prvky lze do globální matice uložit najednou, ne však více- pomocí blokového schématu uvedeného v kapitole 3.4). Použitím efektivního sestavení řídké matice v Matlabu lze tyto přístupy zcela eliminovat a v průběhu jen konstruovat tři vektory, pomocí nichž příkazem v Matlabu řídkou matici seskládáme mnohem rychleji. Tento způsob pracuje tak, že si vytvoříme tři pole  $[vi, vj, vs]$ . Mějme prvek  $p$ , v poli  $vi$  udržujeme informaci o jeho  $i$ . indexu v matici, v poli  $vj$  informaci o  $j$ . indexu v matici a vektor  $vs$  obsahuje samotnou hodnotu. Sestavení pomocí těchto vektorů provedeme příkazem  $K = \text{sparse}(vi, vj, vs, 2 * n, 2 * n)$ . Kromě tří vektorů jako argumentů vidíme ještě dvě hodnoty, ty odpovídají požadovaným rozměrům matice, je totiž lepší uvést toto apriorně, aby metoda *sparse* měla lepší informaci pro konstrukci globální matice. Nyní si ještě můžeme uvést upravený algoritmus sestavení globální matice tuhosti. Hlavní přednost tohoto zápisu tkví pouze v tom, že lokální matici tuhosti mající 36 prvků přečteme jako vektor a každému prvku přiřadíme požadované souřadnice v globální matici tuhosti. Tím přispějeme k nativnímu přístupu Matlabu- k reprezentaci matic pomocí vektorů.

---

**Algorithm 5** Matice tuhosti - efektivní algoritmus
 

---

```

n počet neznámých
for  $i = 1$  to  $n$  do
    vytvoř  $k \in \mathbb{R}^{6 \times 6}$ 
    aktualizuj  $vi, vj, vs$ .
end for
 $K = \text{sparse}(vi, vj, vs, 2 * n, 2 * n)$ 
  
```

---

V Tabulce 5 máme uvedeno, kolik uzlů a kolik elementů přísluší každé úrovni multigridu, kdy iniciační krok byl zvolen  $h = 0.5$ . Konkrétní výsledné hodnoty časů pro sestavení globálních matic tuhosti si ukážeme v následující Tabulce 6. Měření probíhala na notebooku s procesorem Intel Core Duo 1600Mhz. Pro každý čas proběhlo 30 měření a výsledek je průměr naměřených hodnot. Pro lepší názornost jsou tyto hodnoty zaneseny

|          | level 1 | level 2 | level 3 | level 4 | level 5 | level 6 | level 7 | level 8 |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| uzly     | 8       | 25      | 81      | 289     | 1089    | 4225    | 16641   | 66049   |
| elementy | 9       | 32      | 128     | 512     | 2048    | 8129    | 32768   | 131072  |

Tabulka 5: Počet uzlů a elementů pro každou úroveň s iniciálním krokem  $h = 0.5$

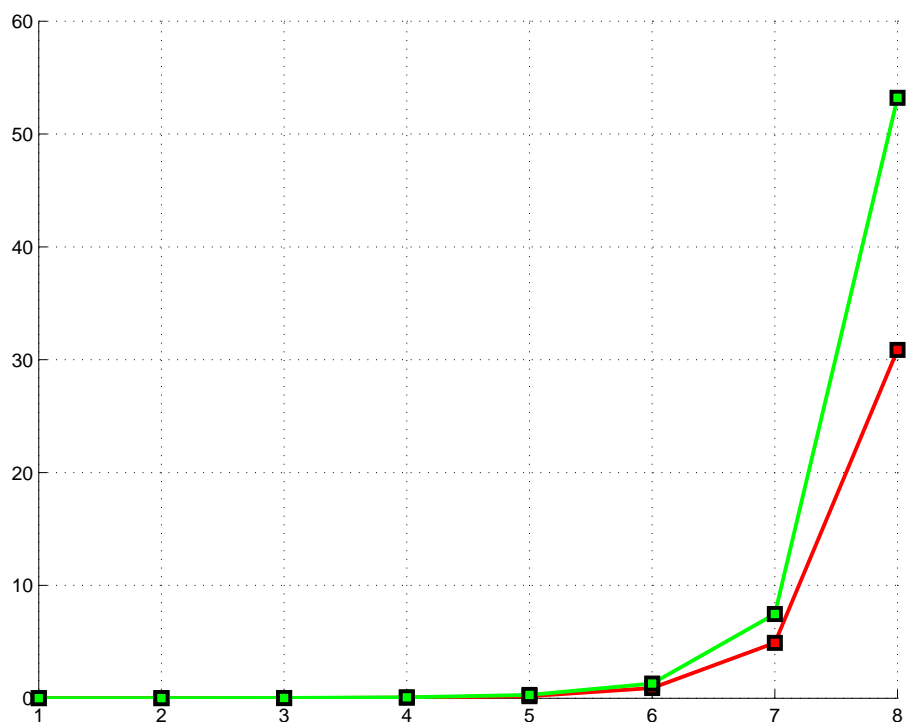
| algoritmus    | level 1 | level 2 | level 3 | level 4 | level 5 | level 6 | level 7 | level 8 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| klasický alg. | 0.0020  | 0.0057  | 0.0190  | 0.0720  | 0.2929  | 1.3009  | 7.4591  | 53.2142 |
| efektivní alg | 0.0017  | 0.0042  | 0.0129  | 0.0455  | 0.1831  | 0.8962  | 4.9093  | 30.8658 |

Tabulka 6: Naměřené časy sestavení matic tuhosti v sekundách

v grafech. Na Obrázku 12 vidíme jak rostou potřebné časy k sestavení globálních matic tuhosti v každém zanoření multigridu. Červeně označený je efektivní algoritmus, zelenou barvou je pak označen klasický přístup konstrukce. Tento graf nám sice ukázal, jak narůstá potřebný čas pro každou úroveň zanoření multigridu a pro oba způsoby implementace, ale podstatnější pro nás bude Tabulka 7 doplněná o graf na Obrázku 13, který ukazuje, jak efektivnější přístup získává výhodu pro sestavení velkých matic. Na Obrázku 13 vidíme, že jsme označili výpočetní čas neupraveného sestavení matice zeleně, což odpovídá 100% a červené hodnoty odpovídají % času z celkových 100%, kterého je potřeba pro sestavení matice optimalizovaným algoritmem. Je dobře vidět nárůst výhody efektivního přístupu. Jinými slovy, je podstatné se zamyslet nad implementací, lze tak ušetřit poměrně dost času při řešení dané metody.

| algoritmus    | level 1 | level 2 | level 3 | level 4 | level 5 | level 6 | level 7 | level 8 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| klasický alg. | 100%    | 100%    | 100%    | 100%    | 100%    | 100%    | 100%    | 100%    |
| efektivní alg | 83%     | 73%     | 68%     | 63%     | 62%     | 69%     | 66%     | 58%     |

Tabulka 7: Procentuální poměr naměřených časů sestavení matic tuhosti

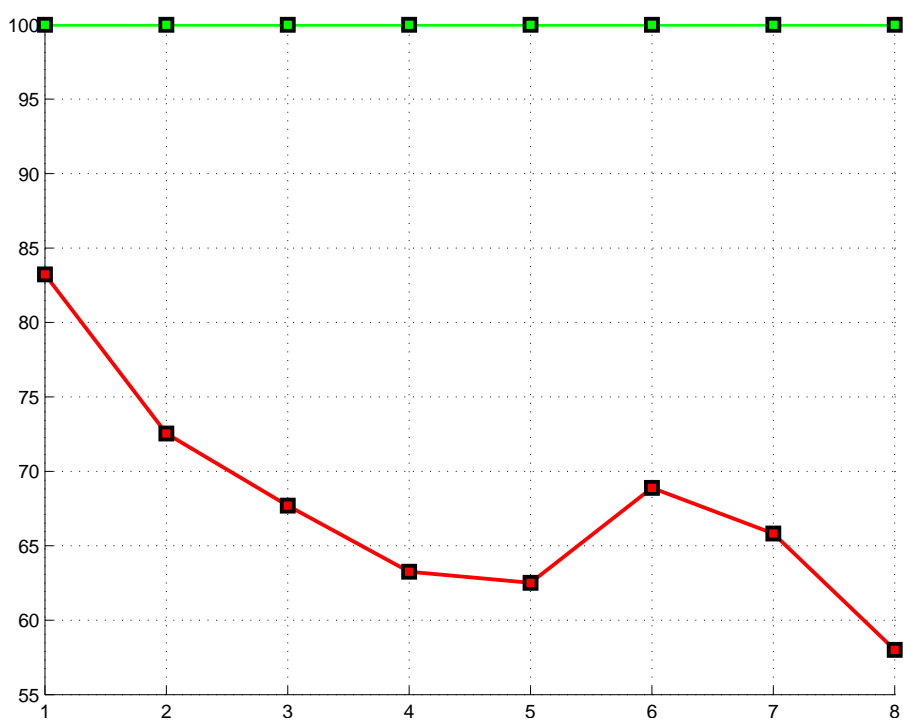


Obrázek 12: Potřebné časy na sestavení matic tuhosti v každé úrovni multigridu

| algoritmus    | level 1 | level 2 | level 3 | level 4 | level 5 | level 6 | level 7 | level 8 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| klasický alg. | 0.0020  | 0.0057  | 0.0188  | 0.0710  | 0.2883  | 1.2797  | 7.4056  | 53.1170 |
| efektivní alg | 0.0016  | 0.0037  | 0.0109  | 0.0379  | 0.1529  | 0.7772  | 4.4530  | 29.2189 |

Tabulka 8: Naměřené časy sestavení matic tuhosti- upravený algoritmus v sekundách

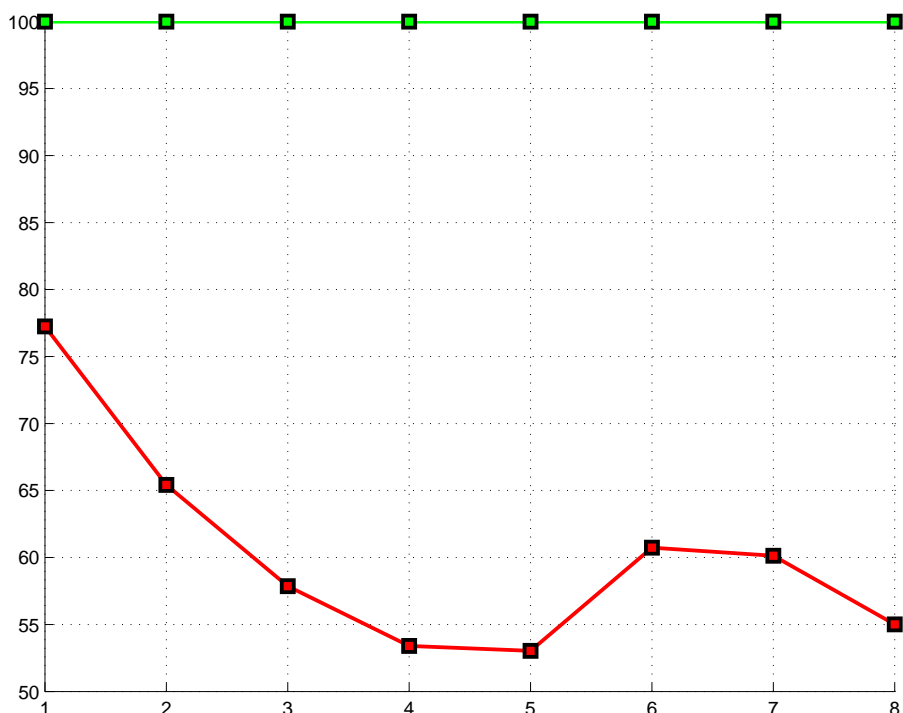
Byla provedena ještě jedna úprava v obou algoritmech. V tomto konkrétním případě došlo k jinému postupu při výpočtu lokální matice tuhosti. Předchozí postup byl vypočten s ohledem na schématický výpočet podle [2], tj. po submaticích. Nynější úprava spočívala v jiném schématu násobení matic, tj. místo násobení po submaticích byla lokální matice tuhosti vypočtena najednou, což postavilo efektivní algoritmus ještě do větší výhody. V klasickém algoritmu se tato úprava projevila jen velmi málo. Ukažme si naměřené hodnoty na následující Tabulce 8 (podmínky měření byly naprosto stejné jako u předchozího měření a změny v algoritmu sestavení byly provedeny v obou případech). Opět v řeči procent si výsledky přepíšeme do Tabulky 9. Výsledek vidíme na Obrázku 14, kde zelenou barvou jsme označili 100% času odpovídající sestavení klasickým algoritmem a červeně pak % času z celkových 100%, kterého je potřeba pro sestavení matice optimalizovaným algoritmem. Vidíme znatelný nárůst výhody takovéto úpravy. V kapitole týkající se tvarové optimalizace zjistíme, že toto je podstatný



Obrázek 13: Procentuální vyjádření potřebného času sestavení matic

| algoritmus    | level 1 | level 2 | level 3 | level 4 | level 5 | level 6 | level 7 | level 8 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| klasický alg. | 100%    | 100%    | 100%    | 100%    | 100%    | 100%    | 100%    | 100%    |
| efektivní alg | 77%     | 65%     | 58%     | 53%     | 53%     | 61%     | 60%     | 55%     |

Tabulka 9: Procentuální poměr naměřených časů sestavení matic tuhosti- upravený algoritmus



Obrázek 14: Procentuální vyjádření potřebného času sestavení matic-úprava

krok, protože takovýchto sestavení matic budeme potřebovat řádově desítky, možná i stovky. Stačí si tedy v řádu těchto čísel vynásobit pouhý rozdíl (což je 1,65 sekund) mezi úpravou efektivního algoritmu a efektivním algoritmem bez úpravy. Bez efektivního přístupu bychom čekali na vyřešení jedné stavové úlohy (v případě největšího zde uvedeného zanoření) o 24 sekund déle, což v následné aplikaci v optimalizační úloze mohou být minuty nebo desítky minut.

#### 4.6.2 Úprava interpolační matice

V průběhu výpočtu multigradní metodou nastane problém při přepočtu reziduí mezi úrovněmi. Tento problém se týká interpolační matice. Konkrétní problém je, že tato matice tak, jak ji korektně zadefinujeme však nerespektuje dirichletovské okrajové podmínky. Při restrikci tedy dojde k tomu, že tam, kde by mělo být ponecháno nulové reziduum plynoucí z okrajových podmínek, tak na tuto pozici jsou zaneseny rezidua z okolních uzlů. Zapišme vše v příkladu, aby tento jev byl zřetelnější.

Nechť máme dvě diskretizace pro level 1 (nejhrubší možná diskretizace) a level 2. Úrovní level 1 odpovídá matici tuhosti  $A_H \in \mathbb{R}^{18 \times 18}$  a aktuální přiblížení řešení  $u_H \in \mathbb{R}^{18}$ , podobně pro level 2 máme k dispozici matici tuhosti  $A_h \in \mathbb{R}^{50 \times 50}$  a aktuální přiblížení řešení  $u_h \in \mathbb{R}^{50}$ . Mějme potom interpolaci  $I \in \mathbb{R}^{50 \times 18}$ , pro restrikci pak tuto matici stačí

transponovat. Pak platí:

$$\begin{aligned} A_h &\approx I A_H, \\ A_H &\approx I^T A_h, \end{aligned}$$

$$\begin{aligned} u_h &= I u_H, \\ u_H &= I^T u_h. \end{aligned}$$

Tyto zápisy jsou obecně v pořádku. V našem případě ale nastal problém při restrikci řešení v multigridní metodě. Jediná nutná změna tkví v tom, že bylo potřeba zohlednit u restrikce dirichletovské uzly. Důvod je ten, že se z okolních uzlů dostávaly nenulové hodnoty při restrikci do uzlů, kam patřila stále nula (s ohledem na dirichletovské uzly vyžadující nulové posuvy). V případě uzlu, kde měla být ponechána nulová hodnota, se jednalo o chybu. Tento jev byl jednoduše řešitelný. V interpolační matici stačilo příslušný řádek danému uzlu, kde jsme chtěli ponechat nulové hodnoty, vynulovat a vložit jedničku na místo příslušné právě tomuto uzlu. V případě nepoužití této úpravy bychom získali zcela nesmyslný výsledek.



## 5 Numerický výpočet stavové úlohy

### 5.1 Příklady diskretizací pro multigradní metodu

V této kapitole se budeme zabývat numerickým řešením konkrétní stavové úlohy. Ukažme si nyní několik možností návrhu diskretizované oblasti. V tomto momentě si zvolíme pevné rozměry:  $K = 1$   $L = 2$   $M = 1$ , takže délka hranic  $\Gamma_1 = 2$ ,  $\Gamma_3 = 1$ ,  $\Gamma_4 = 1$  a  $\Gamma_2$  je reprezentována Bézierovou křivkou s jedním volně umístěným řídicím bodem. Konstanty jsme volili  $E = 10^5 \text{ Nm}^{-2} = 100 \text{ kPa}$  a  $\nu = 0,25$ . Mějme však na paměti, že používáme multigradní numerickou metodu, nebudeme uvádět nejhrubší možnou síť, kterou jsme ostatně viděli dříve na Obrázku 7, protože multigradní metoda využívá ideálně alespoň dvě hierarchické sítě, jinak se ztrácí její význam. Další pozorování je takové, že když začínáme určitou sítí, lze si všimnout například, že když máme oblast rozdělenou na 25 uzlů, která pro potřeby multigradu vychází z předchozí (což je ta, která má 9 uzlů), bude mít hrubší hranici, než síť, kterou bychom přímo vytvořili pro 25 uzlů. Tento jev je dán tím, že vždy, když vytváříme jemnější síť, najdeme vždy střed hrany spojující dva uzly- to je důvod, proč nejsme schopni získat lepší detaily hranice. Na Obrázku 15 vidíme oblast, která je nejhrubší možná pro dvouúrovňový chod multigradní metody s počátečním krokem  $h = 0.5$ .

Síť pro tříúrovňový výpočet vidíme na Obrázku 16.

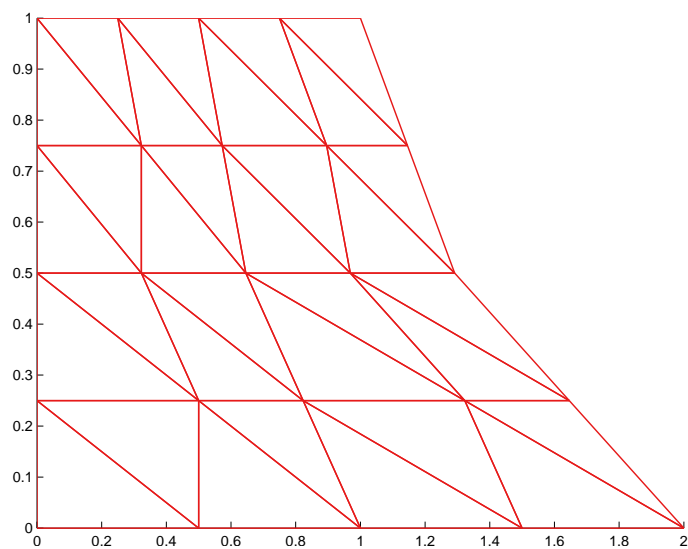
Ukažme si nyní další možnost, s jinou počáteční diskretizací  $h = 0.25$  a  $level = 2$ , ta je na Obrázku 17. Je zřejmé, že máme-li takovou síť pro dvě úrovně, bude mít stejný počet uzlů i elementů jako předchozí síť na Obrázku 16. Tato jinak zvolená síť má již na první pohled hezčí vzhled a lepší detail na hranici.

Pro porovnání si ještě ukažme síť s počáteční diskretizací  $h = 0.25$  pro tři úrovně, která je na Obrázku 18. Potom na Obrázku 19 je síť, která vznikne s počáteční diskretizací  $h = 0.125$  pro dvě úrovně. Je na první pohled patrný větší detail na hranici  $\Gamma_2$ .

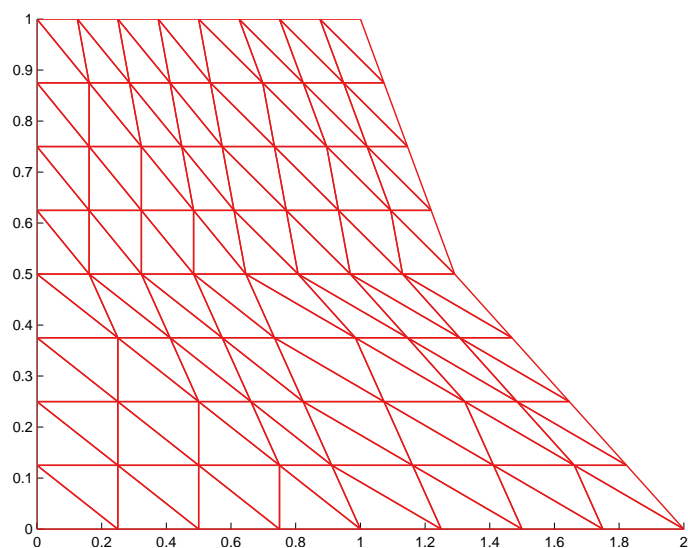
### 5.2 Příklad numerického řešení stavové úlohy

Nyní, když už jsme si přiblížili, jak mohou vypadat diskretizace oblastí a jaké jsou důsledky volby parametru počáteční diskretizace s jedním řídicím bodem. Přejděme nyní k řešení stavové úlohy, čímž získáme posuvy pro jednotlivé uzly.

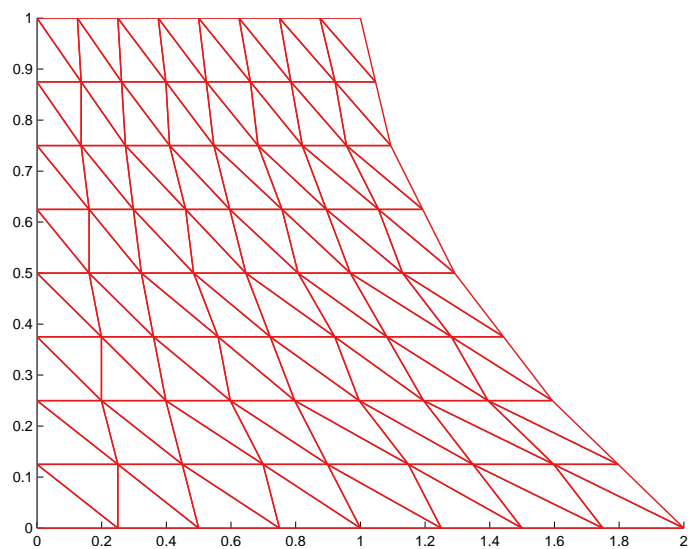
Ukažme si nyní kupříkladu výpočet, kdy jsme použili 15 řídicích bodů pro určení Bézierovy křivky. Volme opět pevné rozměry:  $K = 1$   $L = 2$   $M = 1$ , počáteční diskretizaci  $h = 0.25$  a výpočet provedeme na dvou sítích. Na Obrázku 20 vidíme, jak vypadá zadání. Na Obrázku 21 pak vidíme řešení s naznačenými x složkami napětí a na Obrázku 22 je řešení stejné úlohy, ovšem s y složkami napětí. Dále na Obrázku 23 máme smykovou složku napětí.



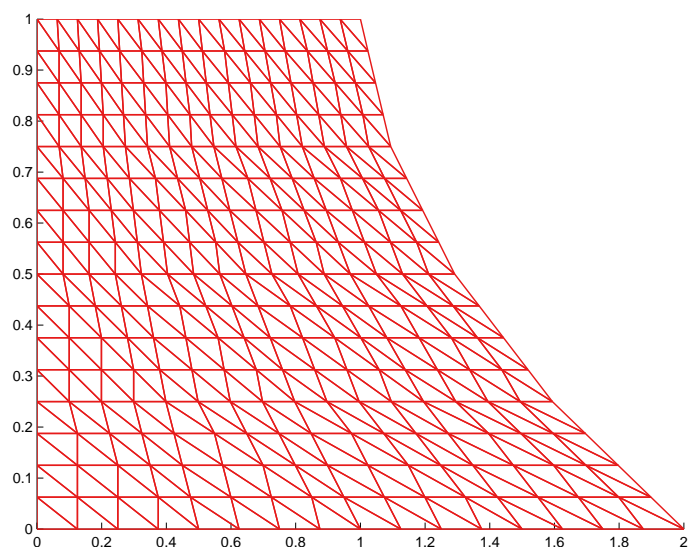
Obrázek 15: Příklad zkoumané oblasti 1



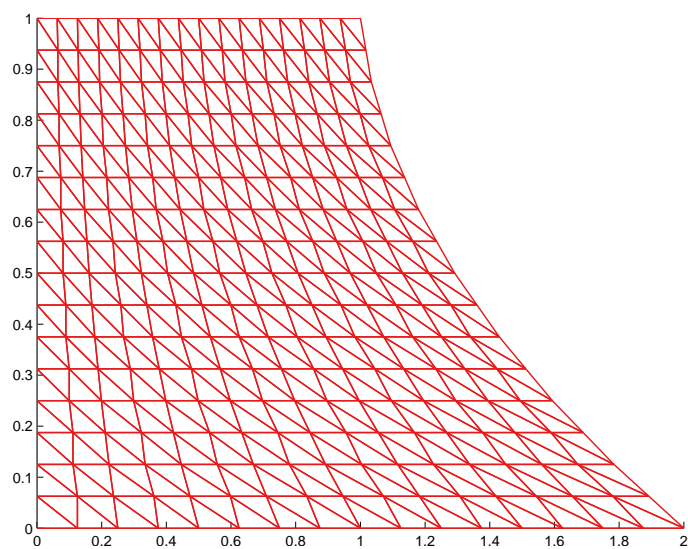
Obrázek 16: Příklad zkoumané oblasti 2



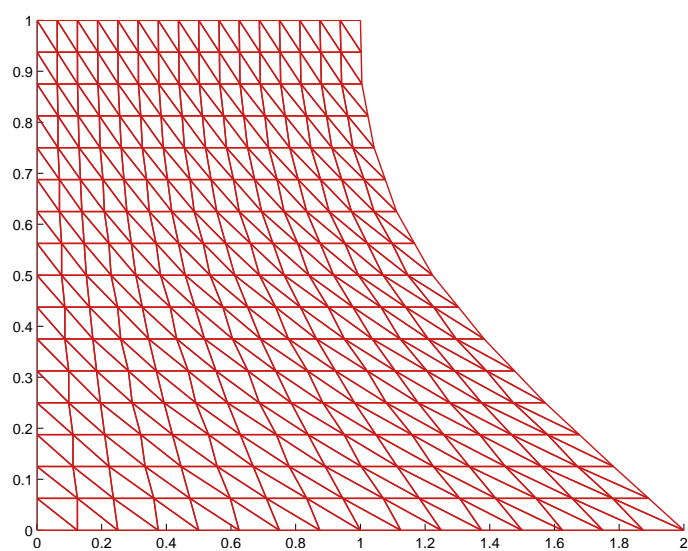
Obrázek 17: Příklad zkoumané oblasti 3



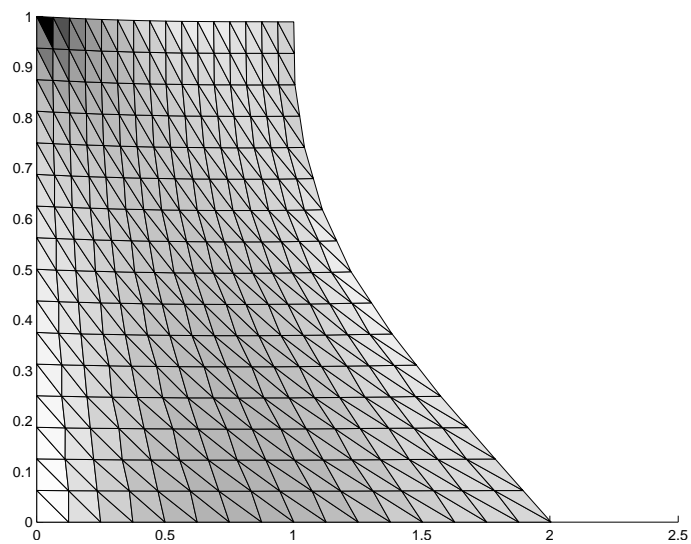
Obrázek 18: Příklad zkoumané oblasti 4



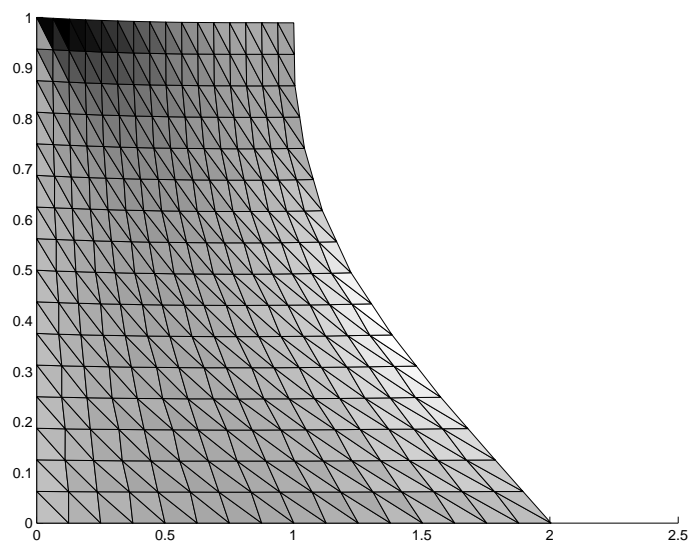
Obrázek 19: Příklad zkoumané oblasti 5



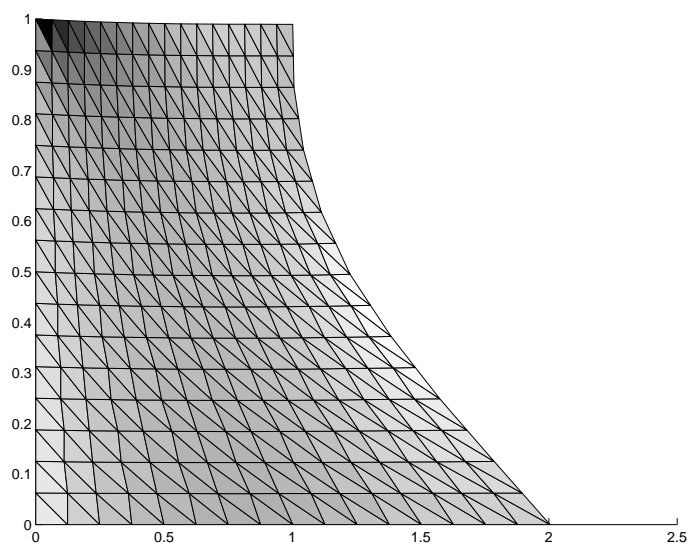
Obrázek 20: Zadání stavové úlohy



Obrázek 21: Řešení stavové úlohy včetně složky  $x$  napětí



Obrázek 22: Řešení stavové úlohy včetně složky  $y$  napětí



Obrázek 23: Řešení stavové úlohy včetně složky smyku napětí

## 6 Formulace tvarově optimalizační úlohy

V této kapitole se budeme zabývat hledáním optimálního tvaru hranice modelovaného Bézierovou křivkou svahu z předchozích kapitol. Jinými slovy nám půjde o hledání optimálního návrhu. Úloha tvarové optimalizace může používat různé cenové funkce vyplývající z určitých požadavků, například následující:

- minimalizace hmotnosti,
- minimalizace napětí,
- maximalizace poddajnosti atd.

Můžeme však současně požadovat kombinace těchto hledisek, které jsou často protichůdné (to znamená, že můžeme požadovat například maximální tuhost a současně minimalizovat hmotnost). V našem případě se budeme zabývat minimalizací napětí na vybraných elementech jejichž množinu indexů si označíme symbolem  $\epsilon$ , přičemž neklademe další omezení pro tuto minimalizaci. Budeme se snažit minimalizaci provádět změnou tvaru Bézierovy křivky a posléze vyhodnocovat napětí, které podrobíme citlivostní analýze. Všechny změny této hranice (a těmto hranicím příslušná kompletní diskretizace) dohromady budou tvořit množinu návrhů.

Z hlediska statiky objektů, jako je námi vyšetřovaný svah, je podstatná volba elementů, které budeme brát v potaz. Můžeme si totiž vybrat ty elementy, které nemusí být právě těmi, kde může dojít k mezním napětím. To je však věcí mnohem hlubší diskuze zacházející již do geologických detailů. Pro nás je však podstatné, že si lze jednoduše předdefinovat množinu elementů, které budeme uvažovat.

### 6.1 Numerické řešení stavového problému

Ukažme si, co vše je součástí řešení stavového problému naší úlohy. Vstupem pro metodu, která řeší stavový problém jsou parametry:

- rozměry  $K, L, M$  (pevně zvolené),
- návrhové body Bézierovy křivky reprezentující hranici  $\Gamma_2$ ,
- počáteční diskretizace,
- počet úrovní zanoření multigradní metody,
- parametry určující elementy, na kterých budeme uvažovat napětí.

Výstupem řešení stavového problému jsou

- posuvy  $u$ ,
- napětí  $\sigma$ ,

kde vektor napětí očekáváme v následujícím tvaru

$$\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix},$$

kde  $\sigma_x$  je napětí ve směru osy  $x_1$ ,  $\sigma_y$  je napětí ve směru osy  $x_2$  a  $\sigma_{xy}$  je smykové napětí.

## 6.2 Tvarová optimalizace

Výše jsme si uvedli, s jakými vstupy a výstupy budeme pracovat. Proces tvarové optimalizace vyžaduje v každém svém optimalizačním kroku vyřešení stavové úlohy. Stavová úloha bude mít pevně zvoleny všechny parametry kromě návrhových bodů Bézierovy křivky- ty budeme v průběhu měnit tak, že v každém optimalizačním kroku provedeme změnu návrhových bodů Bézierovy křivky (a tím se změní i triangulace vyšetřované oblasti). Množinu možných návrhů si označíme symbolem  $U_{ad}$ .  $S$  je zobrazení, které návrhové proměnné  $\alpha$  přiřazuje řešení stavové úlohy  $u, \sigma$ .

Definujme nyní optimalizační úlohu

$$\begin{cases} \min J(\alpha, S(\alpha)) \\ S(\alpha) = (u, \sigma), \alpha \in U_{ad} \end{cases},$$

což můžeme zapsat také takto

$$\min_{\alpha \in U_{ad}} J(\alpha, S(\alpha)).$$

Definujme minimalizační funkcionál  $\Theta$

$$\Theta(\alpha) = J(\alpha, S(\alpha)),$$

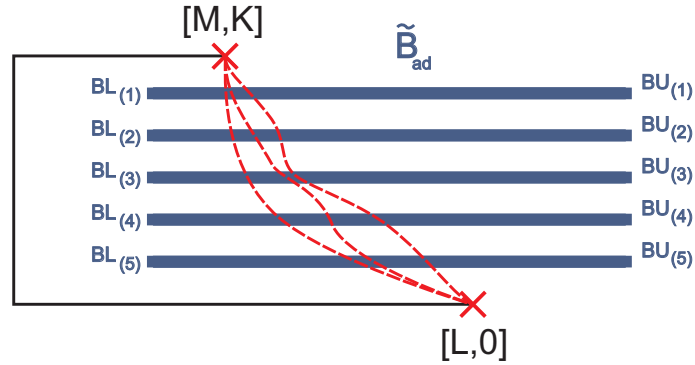
takže budeme minimalizovat funkcionál  $\Theta$  (budeme jej nazývat cenový funkcionál)

$$\min_{\alpha \in U_{ad}} \Theta(\alpha).$$

Nyní, když víme, jakou funkci budeme zkoumat, podívejme se blíže na množinu  $U_{ad}$ . Definujme nejprve rozměry řezu svahu (užitím dříve užívaného značení)  $K = 1, L = 2, M = 1$ . Potom si označme vektor bodů, pomocí nichž bude určena Bézierova křivka

$$\tilde{\alpha} = \begin{bmatrix} [M, K] \\ [\tilde{B}, \tilde{H}] \\ [L, 0] \end{bmatrix} = \begin{bmatrix} [M, K] \\ \alpha \\ [L, 0] \end{bmatrix},$$





Obrázek 24: Možnosti návrhu Bézierovy křivky

kde  $\tilde{B}$  je sloupcový vektor  $p \in \mathbb{N}$  hodnot a  $\tilde{H}$  je sloupcový vektor  $p$  hodnot, je to množina ekvidistantních bodů vypočítaných podle čísla  $p$ . Vektor  $\tilde{B}$  definujeme pomocí dvou vektorů, a to  $BL \in \mathbb{R}^p$  jako vektor omezení zleva a  $BU \in \mathbb{R}^p$  jako vektor omezení zprava následujícím způsobem

$$\tilde{B} = \{x \in \mathbb{R}^p : x_i \in \langle BL_i, BU_i \rangle, i = 1, \dots, p\}.$$

Náš případ množiny, ze které budeme volit tyto body  $\tilde{B}$  a  $\tilde{H}$  je na Obrázku 24. Vidíme modré přímky, odkud budeme volit body  $\tilde{B}$  (v tomto případě pouze schématicky je naznačeno 5 hodnot). Volba  $\tilde{B}$  ( $\tilde{H}$  ponecháváme stejné) z modré oblasti může různým způsobem ovlivnit výsledný tvar Bézierovy křivky, což je například naznačeno třemi červenými přerušovanými čarami představující tři různé návrhy. Pak lze vyjádřit množinu možných návrhových proměnných takto

$$U_{ad} = \left\{ \alpha = (x, y) : x \in \tilde{B} \wedge y \in \tilde{H} \right\}.$$

Nadefinujeme si funkcionál, který budeme mít za úkol minimalizovat takto

$$\Theta(\alpha) = \max_{k \in \epsilon} \left( \sqrt{\sigma_{x_k}^2 + \sigma_{y_k}^2 + \sigma_{xy_k}^2} \right)^2 = \max_{k \in \epsilon} \|\sigma_k\|^2.$$

Nyní budeme hledat minimum funkcionálu  $\Theta$ , k čemuž nám poslouží bundle metoda, o které pojednává následující kapitola.

## 7 Metody hledání minima

Budeme se zabývat numerickými metodami hledání minima funkcí. Nejdůležitější vlastností iteračních metod je možnost jejich realizace na počítači. V této kapitole uvedeme případy hledání minima lipschitzovských funkcí, které nejsou spojitě diferencovatelné v tom smyslu, že množina bodů, v nichž funkce nemá gradient je míry nula. V dalších podkapitolách se budeme zabývat problémem

$$\min_{x \in \Omega} f(x).$$

### 7.1 Pojmy z nehladké analýzy

Definujme nejdříve funkci, jejíž minimum budeme hledat.

**Definice 7.1** *Bud'  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $K > 0$  a množina  $M \subset \mathbb{R}^n$ . Nechť platí  $|f(x) - f(y)| \leq K \|x - y\| \quad \forall x, y \in M$ . Pak řekneme, že  $f$  je lipschitzovsky spojitá na množině  $M$  (s modulem  $K$ ).*

**Definice 7.2** *Řekneme, že  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  je lokálně lipschitzovsky spojitá na okolí bodu  $x$  (s modulem  $K$ ), jestliže pro nějaké  $\epsilon > 0$  je  $f$  lipschitzovsky spojitá s modulem  $K$  na  $x + \epsilon B$ , kde  $B$  je jednotková koule.*

**Definice 7.3** *Řekneme, že  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  je lipschitzovsky spojitá na množině  $M \subset \mathbb{R}^n$ , jestliže je  $f$  lokálně lipschitzovsky spojitá na okolí každého bodu množiny  $M$ .*

**Věta 7.1 (Rademacher)** *Nechť  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  je lokálně lipschitzovsky spojitá na množině  $\mathbb{R}^n$  a nechť  $D = \{x \in \mathbb{R}^n \mid f \text{ je diferencovatelná v } x\}$ . Potom Lebesgueova míra  $\mu$  množiny bodů, ve kterých není funkce  $f$  diferencovatelná je nulová, tzn.  $\mu(\mathbb{R}^n \setminus D) = 0$ .*

**Definice 7.4** *Řekneme, že funkce  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  má v bodě  $x \in \mathbb{R}^n$  jednosměrnou směrovou derivaci ve směru  $h \in \mathbb{R}^n$ , existuje-li konečná limita*

$$f'(x; h) = \lim_{t \downarrow 0} \frac{f(x + th) - f(x)}{t}.$$

**Definice 7.5** *Bud' funkce  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  lokálně lipschitzovsky spojitá v okolí bodu  $x$  a nechť  $h \in \mathbb{R}^n$ , Clarkeova zobecněná směrová derivace funkce  $f$  v bodě  $x$  ve směru  $h$ , kterou značíme  $f^0(x; h)$  je definována předpisem*

$$f^0(x; h) = \lim_{y \rightarrow x} \sup_{t \downarrow 0} \frac{f(y + th) - f(y)}{t},$$

kde  $y \in \mathbb{R}^n$  je vektor a  $t > 0$ .

**Definice 7.6** Buď funkce  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  lokálně lipschitzovsky spojitá v okolí bodu  $x$ . Zobecněný gradient funkce  $f$  v bodě  $x$  s označením  $\partial f(x)$  je množina

$$\partial f(x) = \{ \xi \in \mathbb{R}^n \mid (\xi, h) \leq f^0(x; h) \forall h \in \mathbb{R}^n \},$$

kde elementy  $\partial f(x)$  nazýváme subgradienty funkce  $f$  v bodě  $x$ .

V případě, že funkce  $f$  je spojitě diferencovatelná, pak  $f^0(x; h) = (\nabla f(x), h)$ . Proto  $\partial f(x) = \{ \nabla f(x) \}$ .

**Věta 7.2** Necht  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  je lokálně lipschitzovsky spojitá v okolí bodu  $x$ , potom

$$\partial f(x) = \text{conv} \left\{ \lim_{i \rightarrow \infty} \nabla f(x_i) \mid x_i \rightarrow x, x_i \notin \Omega_f \right\},$$

kde  $\Omega_f = \{x \in \mathbb{R}^n \mid f \text{ není diferencovatelná v } x\}$ .

Následující věta formuluje nutnou podmínku existence minima

**Věta 7.3** Necht  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  je lokálně lipschitzovsky spojitá na množině  $\mathbb{R}^n$  a  $\Omega \subset \mathbb{R}^n$ . Pak platí

$$x^* = \arg \min_{x^k \in \Omega} f(x^k) \Rightarrow 0 \in \partial f(x^*) + N_\Omega(x^*), \forall x^k \in \Omega,$$

kde  $N_\Omega(x^*)$  je normálový kužel v bodě minima.

## 7.2 Subgradientní metoda

Subgradientní metoda je založena na nalezení libovolného subgradientu funkce v daném bodě a aktualizace bodu přiblížení v tomto tvaru

$$\begin{aligned} x^{k+1} &= x^k + t^k d^k, \\ d^k &= -\frac{g^k}{\|g^k\|}, \\ g^k &\in \partial f(x^k). \end{aligned}$$

přičemž požadujeme aby  $t^k \downarrow 0$  a  $\sum_{k=1}^{\infty} t^k = \infty$ , pak  $x^k$  konverguje k optimální hodnotě.

Tato metoda však nezaručuje snížení funkční hodnoty v každém kroku, chybí implementovatelná ukončující podmínka na normu subgradientu a také konvergence je velmi pomalá, menší než lineární. Další problém této metody je ten, že nelze provádět step-length v průběhu výpočtu, protože  $g$  je subgradient a směr  $-g$  nemusí být směr poklesu, takže klasické metody (např. bisekce) pro volbu délky kroku by havarovaly. Musíme volit délku kroku a priori a to lze nejjednodušší například takto

$$t^k = \frac{1}{k}.$$

Taková volba však může být velmi nevhodná v případě, že jsme daleko od minima. Naopak když jsme blízko, snadno můžeme přeskočit hledané minimum.

### 7.3 Bundle metoda

Bundle metoda vychází ze sugradientní metody uvedené výše, přičemž odstraňuje její hlavní nedostatky. Jak z názvu vyplývá, bundle metoda si uchovává tzv. svazek subgradientů- ten postupně vytváří a v jeho konvexním obalu pak hledá nový sestupný směr. Tato metoda zaručuje lepší numerický model subgradientu funkce v daném bodě a také zaručuje sestupný krok pokaždé, když aktualizuje bod přiblížení. U této metody pak také lze hovořit o ukončující podmínce na normu subgradientu.

Iterace bundle metody vypadá následovně

$$\begin{aligned} x^{k+1} &= x^k + t^k d^k, \\ d^k &= -\frac{g^k}{\|g^k\|}, \\ g^k &= \sum_{i=0}^k \lambda_i g^i, g^i \in \partial f(x^i) \forall i \in \mathbb{N}, \\ \sum_{i=0}^k \lambda_i &= 1, \lambda_i \geq 0 \forall i \in \mathbb{N}, \end{aligned}$$

kde  $\lambda_i$  je řešením úlohy kvadratického programování- podrobněji v [8]. Vidíme, že výpočet nového kroku je podobný jako při subgradientní metodě, avšak nevybíráme jeden subgradient, ale konvexní kombinaci subgradientů ze svazku. Uvedme si algoritmus základní bundle metody.

---

**Algorithm 6** Bundle metoda (krok iterace pro výpočet  $x_{k+1}$ )

---

výpočet (sestupného) směru- vektoru  $d^k$

nalezení délky kroku  $t^k > 0$  (steplength)

(a) Serious step - tzn. platí-li  $f(x_{k+1}) < f(x_k)$ , výpočet nového bodu přiblížení  $x_{k+1}$  včetně steplength subrutiny

(b) Null step (je splněno  $f(x_{k+1}) \geq f(x_k)$ ) - provede se  $x_{k+1} = x_k$  a dojde k rozšíření svazku subgradientů o další subgradient  $g_{k+1} \in \partial f(x^k + t^k d^k)$

---

Na základě uvedeného algoritmu snadnou úvahou dojdeme k tomu, že pro realizaci bundle metody budeme potřebovat rutiny pro výpočet funkční hodnoty a libovolného subgradientu cenového funkcionalu. V předchozích kapitolách jsme řešili stavovou úlohu tak, že jsme si navrhli geometrii řezu svahu, nastavili pevně řídicí body Bézierovy křivky reprezentující hranici  $\Gamma_2$  a spustili jsme výpočet. Výstupem pak byly posuvy jednotlivých uzlů v diskretizaci. Nyní jsme před odlišným úkolem. Na řešení této stavové úlohy je potřeba nahlížet jako na funkci zadaných návrhových proměnných stanovujících tvar hranice  $\Gamma_2$  a výstup, který nás bude zajímat, je (jak jsme již dříve uvedli v kapitole 6.2) funkcional závislý na napětích a ten je obecně nespojitě diferencovatelný v závislosti na návrhových proměnných. Takže nyní víme, že funkční hodnotu vypočteme snadno, avšak budeme muset vypočítat také subgradient funkce

(stavové úlohy) v závislosti na návrhových proměnných. Analyticky vypočítat gradient by bylo velmi náročné, zkusíme tedy využít numerického výpočtu subgradientu.

#### 7.4 Výpočet cenového funkcionálu a subgradientu cenového funkcionálu

Výpočet cenového funkcionálu  $\Theta(\alpha) = \Theta(\alpha, u(\alpha), \sigma(\alpha))$ ,  $\alpha \in U_{ad}$ ,  $n \in \mathbb{N}$  bude ve tvaru

$$\Theta(\alpha) = \max_{k \in \epsilon} \left( \sqrt{\sigma_{x_k}^2 + \sigma_{y_k}^2 + \sigma_{xy_k}^2} \right)^2 = \max_{k \in \epsilon} \|\sigma_k\|^2, \quad \alpha \in U_{ad}.$$

Všimněme si, že cenový funkcionál je závislý pouze na napětích. Subgradient cenového funkcionálu vypočteme následovně

$$\partial\Theta(\alpha) \ni \tilde{\nabla}_{\alpha}\Theta(\alpha) = \nabla_{\alpha}\Theta(\alpha, u(\alpha), \sigma(\alpha)) = \begin{bmatrix} \frac{\partial\Theta(\alpha)}{\partial\alpha_1} \\ \vdots \\ \frac{\partial\Theta(\alpha)}{\partial\alpha_n} \end{bmatrix}, \quad \alpha \in U_{ad} \quad i = 1 \dots n,$$

kde

$$\frac{\partial\Theta(\alpha)}{\partial\alpha_i} \approx \frac{\Theta(\alpha + \epsilon[0, \dots, 0, 1_i, 0, \dots, 0]) - \Theta(\alpha)}{\epsilon}.$$

#### 7.5 Bundle trust metoda

Pro řešení naší optimalizační úlohy jsme použili bundle trust metodu. Bundle trust metoda je kombinace bundle metody a trust region metody. Mezi bundle trust metodou a bundle metodou je zásadní rozdíl v tom, že u bundle trust metody místo kroku step-length provádíme kontrolu, jak dobře odpovídá model cenovému funkcionálu a poté buď zvětšíme nebo zmenšíme region důvěry. Kritéria pro zvětšení či zmenšení regionu důvěry jsou dány určitou heuristikou a v našem případě se jimi zabývat nebudeme. Pro minimalizaci našeho cenového funkcionálu použijeme existující implementaci, k níž potřebné informace lze získat z [7], implementace této metody přesahuje rámec této práce, proto byl použit existující kód s názvem BTNCLC- implementace ve Fortranu a spuštění v Matlabu pomocí mex rozhraní. BTNCLC požaduje určité parametry pro spuštění a také metody pro výpočet funkční hodnoty cenového funkcionálu a subgradientu cenového funkcionálu. Vypišme si nejprve pro nás podstatné parametry tak, jak jsou značeny v rámci BTNCLC (v závorce je naše označení)

- $N(N)$ - počet proměnných, podle kterých provádíme optimalizaci,
- $X(\alpha)$ - vektor  $N$  hodnot, podle kterých bude program provádět optimalizaci (odpovídá to proměnným, přes které budeme numericky počítat subgradient),
- $BL(BL)$ - vektor  $N$  hodnot omezující  $X$  zleva,
- $BU(BU)$ - vektor  $N$  hodnot omezující  $X$  zprava,

- FM (*FM*)- odhad minima funkce,
- EPS (*EPS*)- přesnost pro řešení nalezení minima bundle trust metodou,
- MAXCOM (*MAXCOM*)- maximální počet vyčíslení cenového funkcionálu,
- MAXIT (*MAXIT*)- maximální počet iterací algoritmu bundle trust metody,
- RESET (*RESET*)- maximální počet subgradientů ve svazku, po překročení dojde k resetu svazku,
- IPRINT (*IPRINT*)- hodnota pro určení parametrů pro výpis výsledků optimalizačního procesu do souboru.

Ostatní parametry se týkají omezení na rovnost a nerovnost, tyto jsme nevyužili.

## 8 Experimenty s optimalizační úlohou

Uvedeme si nyní několik příkladů optimalizační úlohy. Rozměry řezu svahu budou ve všech případech  $L = 2$ ,  $M = 1$ ,  $K = 1$  a volíme  $E = 10^5$  a  $\nu = 0,25$ . Další parametry v seznamu byly pevně zvoleny tak, abychom mohli zkoumat pouze vliv změn návrhových bodů.

- FM (*FM*)-  $3 \cdot 10^2$
- EPS (*EPS*)-  $10^{-2}$
- MAXCOM (*MAXCOM*)- 1000
- MAXIT (*MAXIT*)- 1000
- RESET (*RESET*)- 80
- IPRINT (*IPRINT*)- 4

### 8.1 Příklad 1

Zadání:

Hranici  $\Gamma_2$ , kterou budeme optimalizovat bude představovat 15 řídících bodů pro Bézierovu křivku. Definujme iniciální hodnoty

$$\tilde{H} = \begin{bmatrix} 0.9375, 0.875, 0.8125, 0.75, 0.6875, 0.625, 0.5625, 0.5, \dots \\ 0.4375, 0.3750, 0.3125, 0.25, 0.1875, 0.125, 0.0625 \end{bmatrix},$$

$$\tilde{B} = \begin{bmatrix} 1.0625, 1.125, 1.1875, 1.25, 1.3125, 1.375, 1.4375, 1.5, \dots \\ 1.5625, 1.625, 1.6875, 1.75, 1.8125, 1.875, 1.9375 \end{bmatrix}.$$

Budeme hledat s počátečními hodnotami  $\alpha = [\tilde{B}, \tilde{H}]$  optimální vektor  $\alpha_{opt}$ , pro ten však potřebujeme omezení ve formě dříve zmíněných vektorů  $BL$  a  $BU$ , protože můžeme najít jistá lokální minima- avšak nelze s jistotou říci, že je nalezeno globální minimum. Proto je potřeba provádět mnoho testů a zvažovat, jak který návrh odpovídá realitě. Potom jako vektor  $BL$  omezující zleva vypočtené hodnoty (tedy hodnoty pro optimalizaci) bude ve tvaru

$$BL = \begin{bmatrix} 0.225, 0.35, 0.3875, 0.42, 0.5125, 0.615, 0.7375, 0.8, \dots \\ 0.8625, 0.925, 1.0875, 1.15, 1.2125, 1.275, 1.3375 \end{bmatrix},$$

vektor  $BU$  omezující zprava hodnoty výsledného vektoru

$$BU = \begin{bmatrix} 1.3625, 1.425, 1.4875, 1.55, 1.6125, 1.675, 1.7375, 1.8, \dots \\ 1.8625, 1.925, 2.0875, 2.15, 2.2125, 2.275, 2.3375 \end{bmatrix}.$$

Nyní si ukažme, jak vypadá situace, kterou máme zadanou. Na Obrázku 25 vidíme 4 obrázky (označme si je čísla podle pořadí od prvního):

1. napětí- složka  $x$
2. napětí- složka  $y$
3. napětí- složka smyku
4. řešení pro iniciální návrh

Poznamenejme, že první tři obrázky se liší pouze zobrazenými napětími (nutno dodat, že napětí jsou ve třech složkách řádově jiné velikosti, ale pro lepší zobrazení došlo v každém obrázku ke znormování v příslušné složce na interval  $\langle 0, 1 \rangle$ ). Tyto obrázky ukazují rovněž, jak vypadal počáteční návrh a to bez deformací, pro lepší porovnání s optimalizovaným i ostatními návrhy. Dále je možno si všimnout, že nebereme v potaz pouze napětí na hranici  $\Gamma_2$ , ale i na hranici  $\Gamma_3$  a to proto, že na  $\Gamma_3$  hranici mohou vznikat také poměrně velká napětí vlivem optimalizačního procesu, což také potvrzují vykreslené obrázky.

Výstupem optimalizačního procesu je  $\alpha_{opt}$  s optimalizovanými hodnotami  $\tilde{B}$  ( $\tilde{H}$  jsme v rámci optimalizačního procesu neměnili)

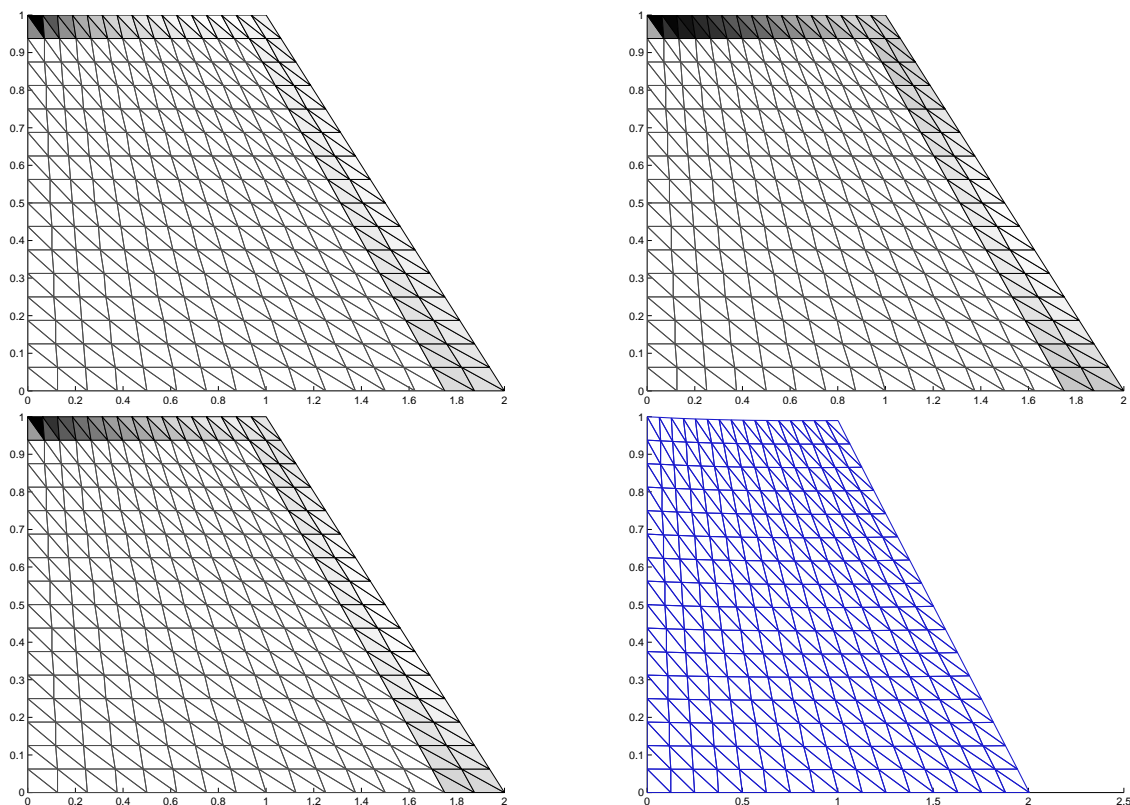
$$\tilde{B} = \begin{bmatrix} 0.8263, 0.7311, 0.5905, 0.5287, 0.5387, 0.615, 0.7375, 0.8, \dots \\ 0.8625, 0.925, 1.0875, 1.15, 1.7421, 2.275, 2.3375 \end{bmatrix}.$$

Pro tento optimalizovaný návrh vidíme výsledek na Obrázku 26, kde vidíme opět 4 obrázky (ve stejném pořadí jednotlivých obrázků, jako v neoptimalizovaném tvaru): Podstatné je pro nás to, jak se změnil cenový funkcionál- původně vyčíslený cenový funkcionál je 46 krát větší než ten, který jsme získali po optimalizaci, což je podstatně velká změna. Můžeme si také všimnout následující tabulky 10, kde jsou vypsány optimalizační kroky. Je zřetelné, jak se snižoval cenový funkcionál v průběhu optimalizačního procesu.



| iterace | počet vyčíslení $\Theta$ | funkční hodnota $\Theta$ |
|---------|--------------------------|--------------------------|
| 1       | 1                        | $0.4202 \cdot 10^5$      |
| 2       | 7                        | $0.2095 \cdot 10^5$      |
| 3       | 8                        | $0.1386 \cdot 10^5$      |
| 4       | 9                        | $0.6708 \cdot 10^4$      |
| 5       | 10                       | $0.3665 \cdot 10^4$      |
| 6       | 11                       | $0.3665 \cdot 10^4$      |
| 7       | 12                       | $0.2798 \cdot 10^4$      |
| reset   |                          |                          |
| 8       | 13                       | $0.1802 \cdot 10^4$      |
| 9       | 14                       | $0.1205 \cdot 10^4$      |
| 10      | 16                       | $0.1121 \cdot 10^4$      |
| 11      | 17                       | $0.9375 \cdot 10^3$      |
| 12      | 19                       | $0.9215 \cdot 10^3$      |
| 13      | 20                       | $0.9109 \cdot 10^3$      |
| 14      | 23                       | $0.9109 \cdot 10^3$      |

Tabulka 10: Příklad 1- výsledky optimalizačního procesu

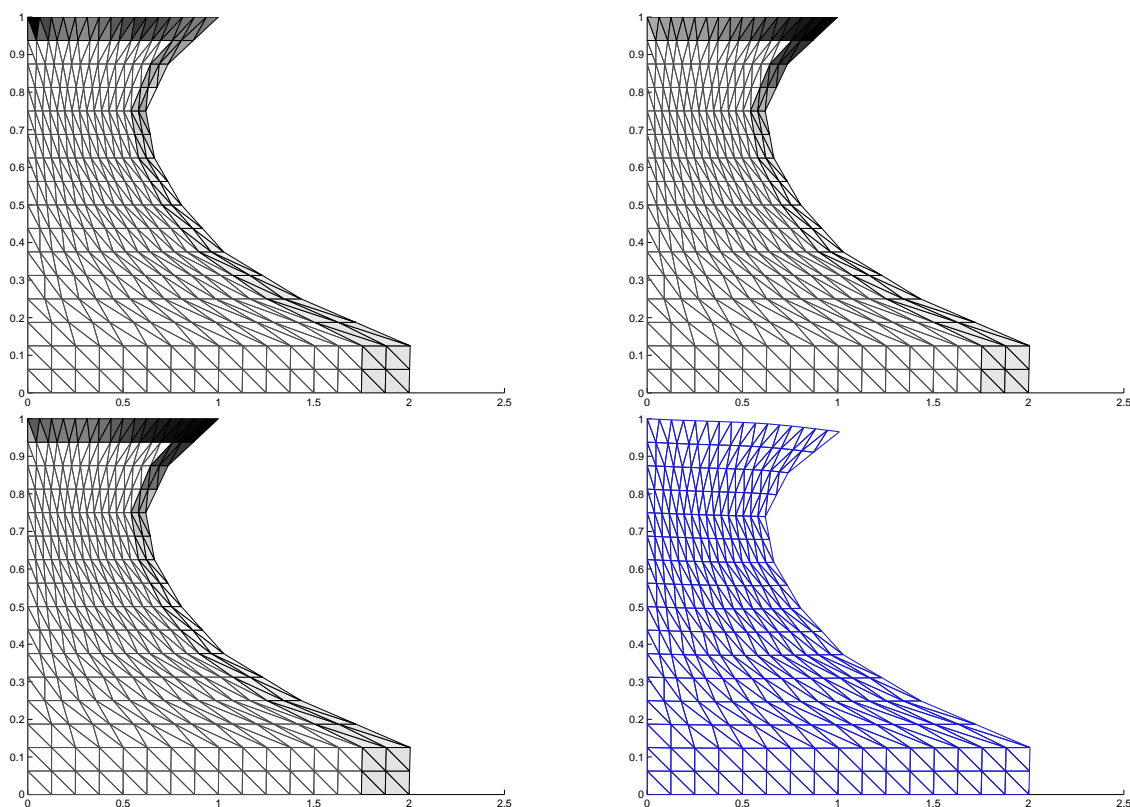


Obrázek 25: Před optimalizací (Př. 1). Napětí- složka x; složka y; složka smyk; Řešení

## 8.2 Příklad 1- zkoumání závislosti cenové funkce na změně jednotlivých složek návrhové proměnné

V tomto momentě potřebujeme zjistit, jak dobrý jsme dostali vektor návrhových proměnných. Výsledky analýzy vidíme na Obrázku 27, kde je uvedeno celkem 15 grafů. Každý z nich odpovídá jedné proměnné již optimalizovaného vektoru  $\tilde{B}$ , který nyní symbolicky přeznačíme takto  $\tilde{B} = [x_1, \dots, x_{15}]$  a poznamenejme, že každá proměnná  $x_i$  v příslušném grafu je zvýrazněna červenou hvězdičkou. Pro každou proměnnou provedeme následující proces:

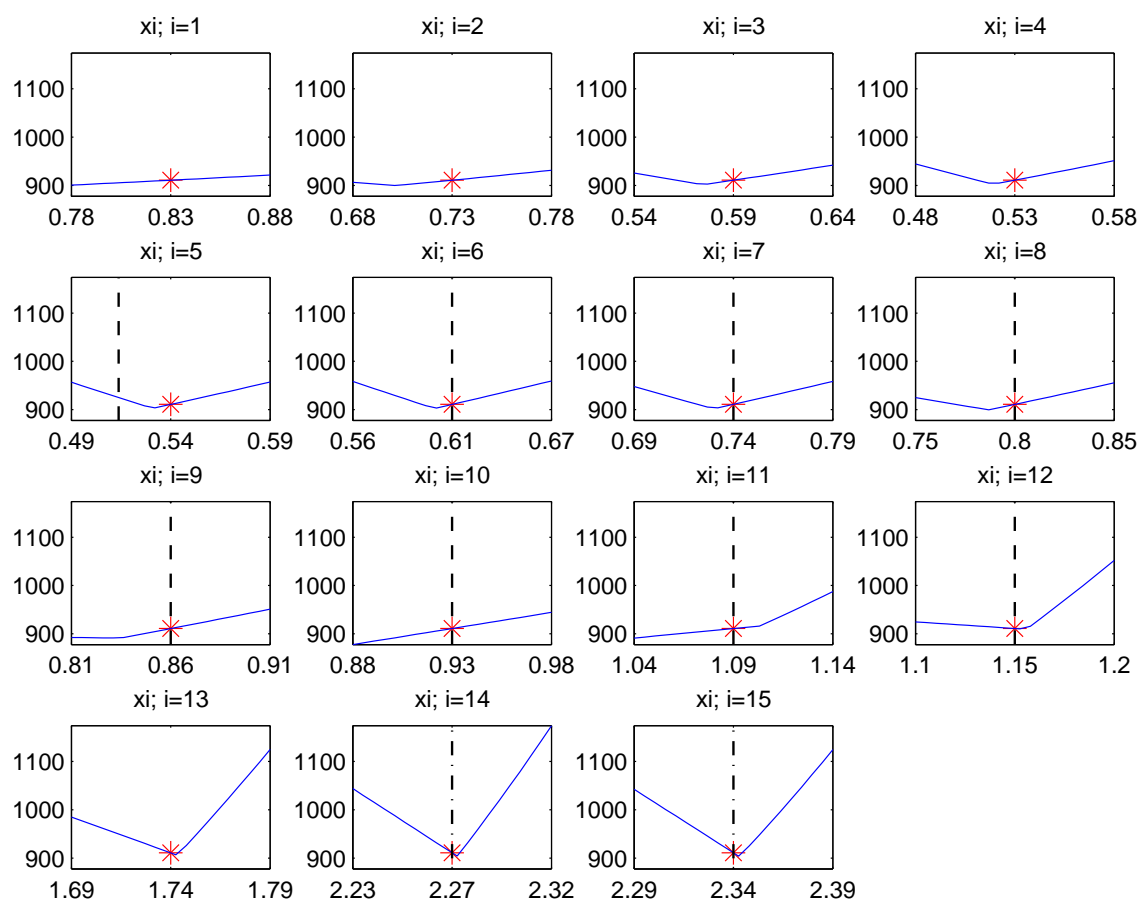
1. definujeme  $eps > 0$
2. vybereme požadovanou proměnnou  $x_i$ ,  $i = 1, \dots, 15$  a ostatní ponecháme zafixovány
3. definujeme interval  $I$ ,  $I = \langle x_i - eps, x_i + eps \rangle$
4. provedeme v cyklu  $n$  krát výpočet stavové úlohy v závislosti na  $x_i \in I$ , tím dostaneme funkci jedné proměnné  $x_i$  s funkční hodnotou  $\Theta$ .



Obrázek 26: Po optimalizaci (Př. 1). Napětí- složka x; složka y; složka smyk; Řešení

5. vykreslíme tento graf modrou barvou s oborem hodnot  $I$  a tímto způsobem zpracujeme všechny zbývající proměnné
6. bod nalezený optimalizačním procesem označíme červenou hvězdičkou
7. je-li přítomno omezení BL v oboru hodnot grafu, vykreslíme jej přerušovanou černou čarou
8. je-li přítomno omezení BU v oboru hodnot grafu, vykreslíme jej čerchovanou černou čarou

Výsledek analýzy by měl být ideálně pro každou proměnnou ve formě ryze konvexní funkce, kde  $x_i$  je v minimu případně konstantní funkce. V našem příkladě se tak ve všech případech nestalo, to však nemusí být nutně chyba, ale jen známka toho, že jsme byli příliš přísní při nastavování vektorů  $BL$  a  $BU$ . V 1. proměnné je vidět velmi malá změna funkční hodnoty. Ve 2., 3. a 4. proměnné najdeme také malou změnu funkční hodnoty a lze si všimnout, že nalezené body jsou blízko minima této funkce. U 5. proměnné již je vidět i omezení, avšak řešení tím není ovlivněno, opět je nalezen bod téměř u minima. Pro 6. až 12. proměnnou pozorujeme omezení zleva vektorem  $BL$  a přesto jsme také velmi blízko minima, kromě 10. a 11. proměnné, kde funkční hodnota ještě dále klesá.



Obrázek 27: Analýza citlivosti pro každou proměnnou z příkladu 1

Pro 13. proměnnou zase vidíme korektně nalezené minimum a omezení jsou dostatečně daleko od tohoto bodu. Pro 14. a 15. proměnnou jsme narazili také na omezení, avšak už je to omezení vektorem  $BU$ , které je také velmi blízko minima. Proto lze tvrdit, že omezení byla volena rozumně. Ačkoliv nebylo nalezeno vždy přesné minimum (naše úloha je poměrně náročná), u všech proměnných bylo dosaženo dobrých výsledků. Tuto optimalizaci tedy lze označit za zdařilou. Lze si také všimnout, že naše grafy jsou zjevně nespojitě diferencovatelné, čímž se potvrzuje, že funkcionál  $\Theta$  je opravdu nehladký.

### 8.3 Příklad 2

V tomto příkladě bude situace naprosto stejná, jako v předchozím případě, ale s tím, že změníme tři pole. Pole  $\tilde{H}$  je stejné, jako v předchozím příkladě. Změníme pole  $\tilde{B}$ ,  $BL$ ,  $BU$  takto

$$\begin{aligned}\tilde{B} &= \begin{bmatrix} 1.11, 1.23, 1.31, 1.42, 1.51, 1.6, 1.67, 1.71, \dots \\ 1.74, 1.75, 1.76, 1.77, 1.78, 1.82, 1.93 \end{bmatrix} \\ BL &= \begin{bmatrix} 0.99, 1.0, 1.01, 1.03, 1.06, 1.09, 1.14, 1.19, \dots \\ 1.25, 1.36, 1.44, 1.54, 1.62, 1.7, 1.75 \end{bmatrix}, \\ BU &= \begin{bmatrix} 1.12, 1.24, 1.33, 1.44, 1.53, 1.61, 1.69, 1.73, \dots \\ 1.75, 1.76, 1.77, 1.82, 1.84, 1.85, 1.94 \end{bmatrix}.\end{aligned}$$

Je možno si povšimnout toho, že jsme mnohem více omezili  $BL$  a  $BU$ , tak, jak bychom mohli intuitivněji odhadnout výsledek. Nyní si ukažme, jak vypadá situace, kterou máme zadanou. Na Obrázku 28 vidíme 4 obrázky (označme si je čísla podle pořadí od prvního):

1. napětí- složka x
2. napětí- složka y
3. napětí- složka smyku
4. řešení pro iniciální návrh

Výstupem optimalizačního procesu je  $\alpha_{opt}$  s optimalizovanými hodnotami  $\tilde{B}$  ( $\tilde{H}$  jsme v rámci optimalizačního procesu opět neměnili)

$$\tilde{B} = \begin{bmatrix} 0.9900, 1.0000, 1.0100, 1.0300, 1.0600, 1.0900, 1.1400, 1.1900, \dots \\ 1.2500, 1.4685, 1.7700, 1.8200, 1.8400, 1.8500, 1.9400 \end{bmatrix}.$$

Pro tento návrh vidíme výsledek na Obrázku 29 ve stejném pořadí jednotlivých obrázků, jako v neoptimalizovaném tvaru.

Všimněme si znova, jak se změnil cenový funkcionál- původně vyčíslený cenový funkcionál je 15 krát větší než ten, který jsme získali po optimalizaci. V tabulce 11 je opět výstup optimalizačního procesu, kde jsou naznačené optimalizační kroky. Opět si povšimněme, jak se měnilo vyčíslení cenového funkcionálu.

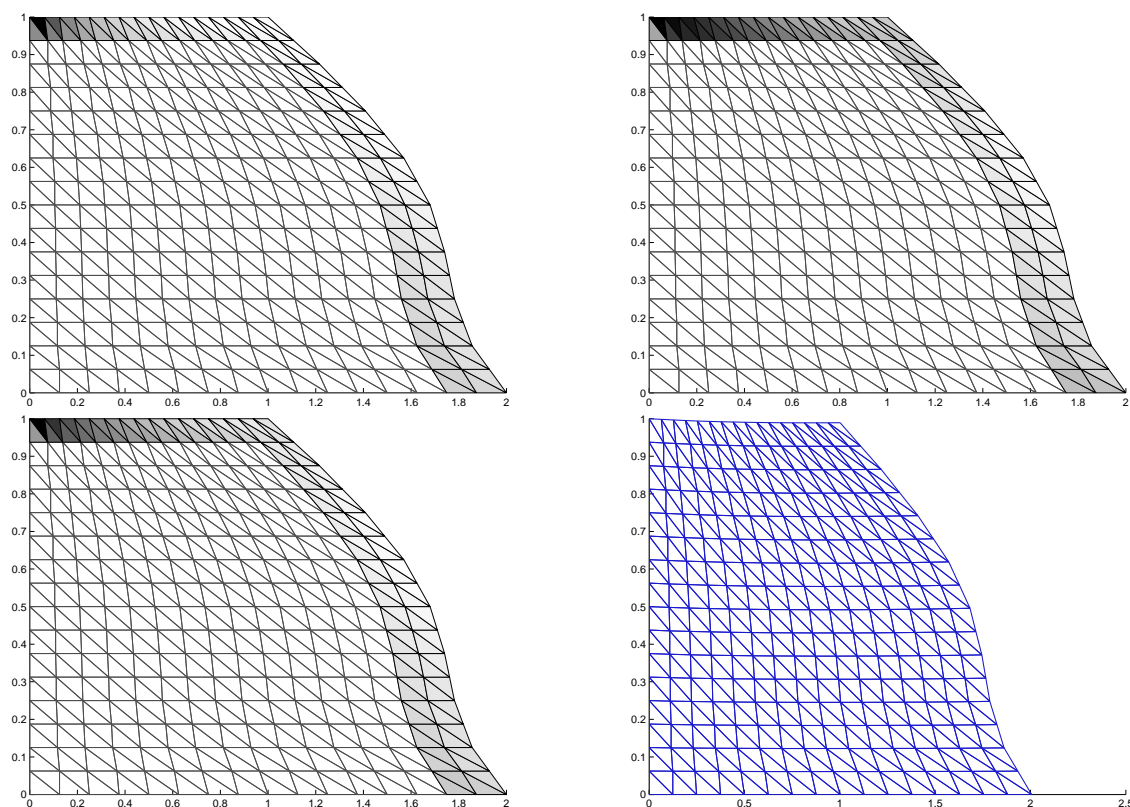
| iterace | počet vyčíslení $\Theta$ | funkční hodnota $\Theta$ |
|---------|--------------------------|--------------------------|
| 1       | 1                        | $0.2027263 \cdot 10^6$   |
| 2       | 2                        | $0.2304439 \cdot 10^5$   |
| 3       | 3                        | $0.1543313 \cdot 10^5$   |
| 4       | 4                        | $0.1354244 \cdot 10^5$   |
| 5       | 9                        | $0.1336706 \cdot 10^5$   |
| 6       | 10                       | $0.1327372 \cdot 10^5$   |
| 7       | 11                       | $0.1327327 \cdot 10^5$   |
| reset   |                          |                          |
| 8       | 24                       | $0.1327327 \cdot 10^5$   |
| 9       | 25                       | $0.1327324 \cdot 10^5$   |

Tabulka 11: Příklad 2- výsledky optimalizačního procesu

#### 8.4 Příklad 2- zkoumání závislosti cenové funkce na změně jednotlivých složek návrhové proměnné

Pro druhý příklad provedeme stejný proces analýzy jako v příkladě 1 i se stejným značením. Na Obrázku 30 vidíme výsledky analýzy. Výsledek dopadl dle očekávání. Tento příklad jsme totiž více omezili pomocí polí  $BL$  a  $BU$ . Pro první 3 proměnné pozorujeme omezení zleva, nalezený bod je blízko těchto omezení  $BL$  a také pozorujeme blízké minimum funkcí vpravo od nalezených bodů. Jedná se však o malý rozdíl cenového funkcionálu  $\Theta$ . Vlivem většího omezení  $BL$  jsou proměnné 4. až 9. zřejmě více omezeny a optimalizačním procesem nalezené body jsou právě v tomto bodě omezení. V 10. proměnné je nalezen bod téměř v minimu a není zde blízko omezení, kdežto u 11. a 12. vidíme omezení zprava, ale přesto je vpravo od požadovaného bodu minima. V proměnných 13. až 15. máme omezení zprava a je vidět, že by cenový funkcionál  $\Theta$  mohl být ještě menší. Toto omezení však ale můžeme chápat jako požadavek většího množství vytěženého materiálu.

Nyní, když jsme pracně vše implementovali a odzkoušeli a také viděli teoretické výpočty, podívejme se, jak vypadá takovýto svah doopravdy. Můžeme takto porovnat naše výstupy a realitu. Je ovšem jasné, že náš model je pouze lineární a nebere v potaz pórovitost, vlhkost a další reálné součinitele. Na Obrázku 31 vidíme svah vzniknuvší po zásahu korečkovým rypadlem. Tento fragment je vyfocen v těžebním Lomu Bílina na severo-západě Čech.



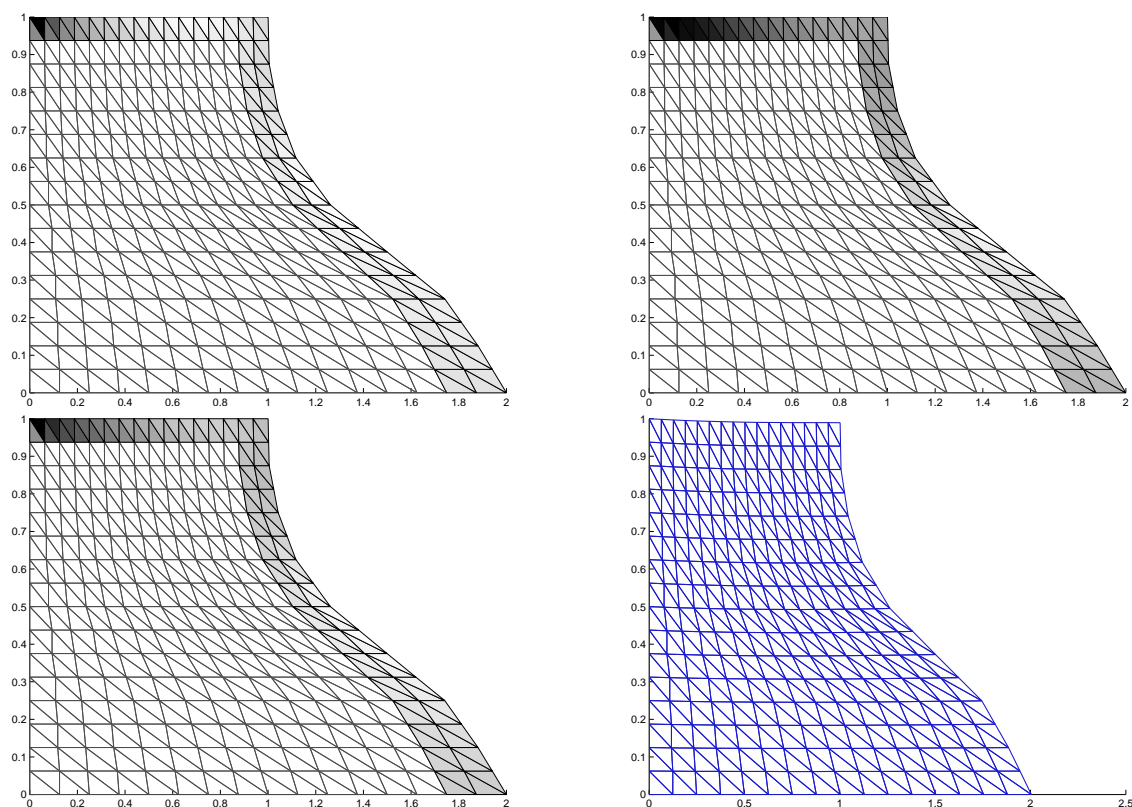
Obrázek 28: Před optimalizací (Př. 2). Napětí- složka x; složka y; složka smyk; Řešení

## 8.5 Grafické uživatelské rozhraní

Jako součást této práce bylo vytvořeno grafické uživatelské rozhraní pro jednodušší manipulaci i zobrazení zadání a výsledků optimalizačního procesu. Tento program je jednoduše spustitelný příkazem GUI v Matlabu. Jeho zobrazení včetně popisu vidíme na Obrázku 32. Pomocí tohoto programu je možno si nechat vypočítat zde uvedené příklady, případně ještě upravené varianty. Zadávat lze počet zanoření multigridu z nejhrubší sítě polem *level*. Pole *h* je diskretizace nejhrubší sítě (resp. hranice  $\Gamma_4$ , kde zbytek oblasti je přizpůsoben zadané geometrii). Pole *posun* definuje, které elementy chceme brát v úvahu, tj. zadáním  $-1$  bereme v potaz všechny elementy,  $0$  definuje pouze krajní elementy na hranicích  $\Gamma_2$  a  $\Gamma_3$  a čím je toto číslo vyšší, tím hlouběji budeme uvažovat napětí na elementech (nás však spíše zajímají napětí blízko hranice  $\Gamma_2$ ). Dále vidíme pole *B*- zadání počátečních řídicích bodů, *BL*- dolní omezení pro výsledek optimalizace, *BU*- horní omezení pro výsledek optimalizace, *Bopt*- výsledný vektor řídicích bodů optimalizace a všechny tyto 4 vektory musí mít stejnou délku- v našem případě 15.

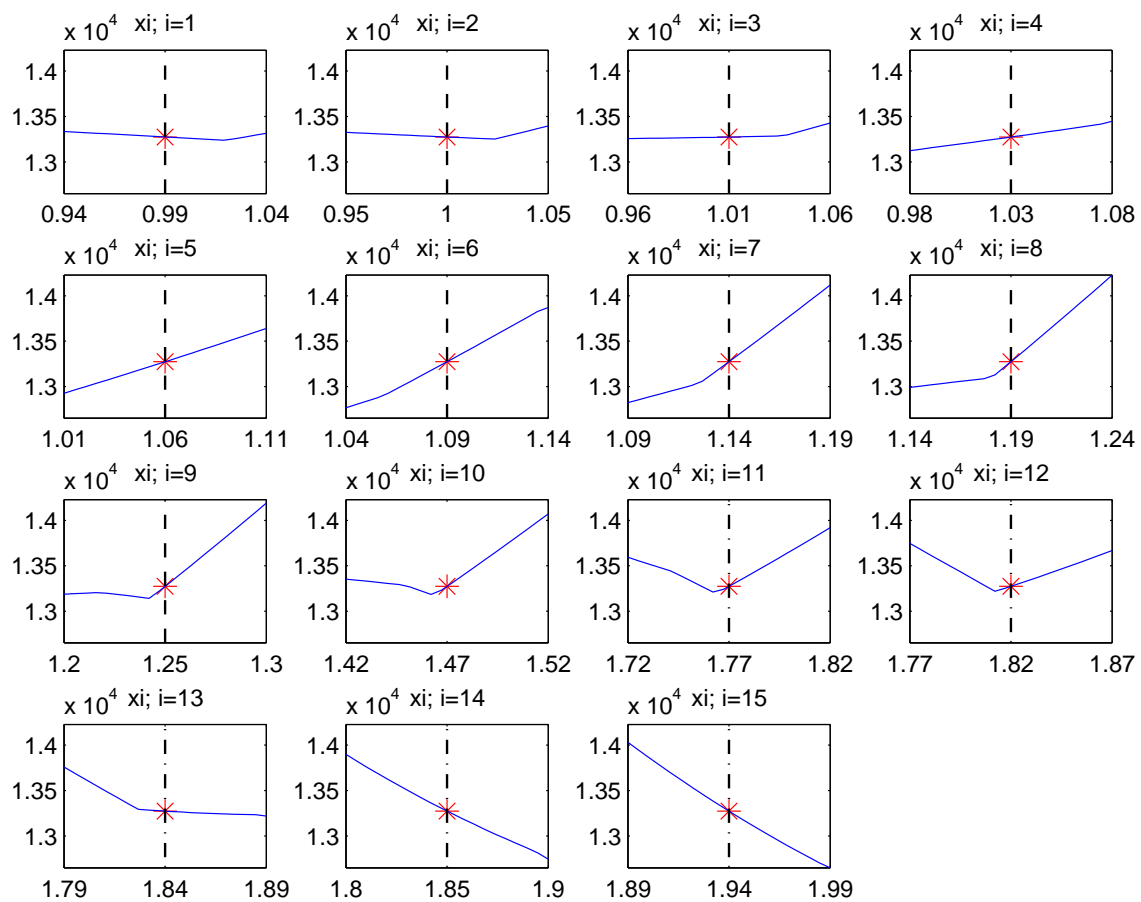
Vidíme také dvě tlačítka, pomocí tlačítka *optimalizuj* provedeme optimalizační proces. Po tomto výpočtu lze zjistit, jak dobré řešení bylo nalezeno pomocí tlačítka *citlivost*.





Obrázek 29: Po optimalizaci (Př. 2). Napětí- složka x; složka y; složka smyk; Řešení

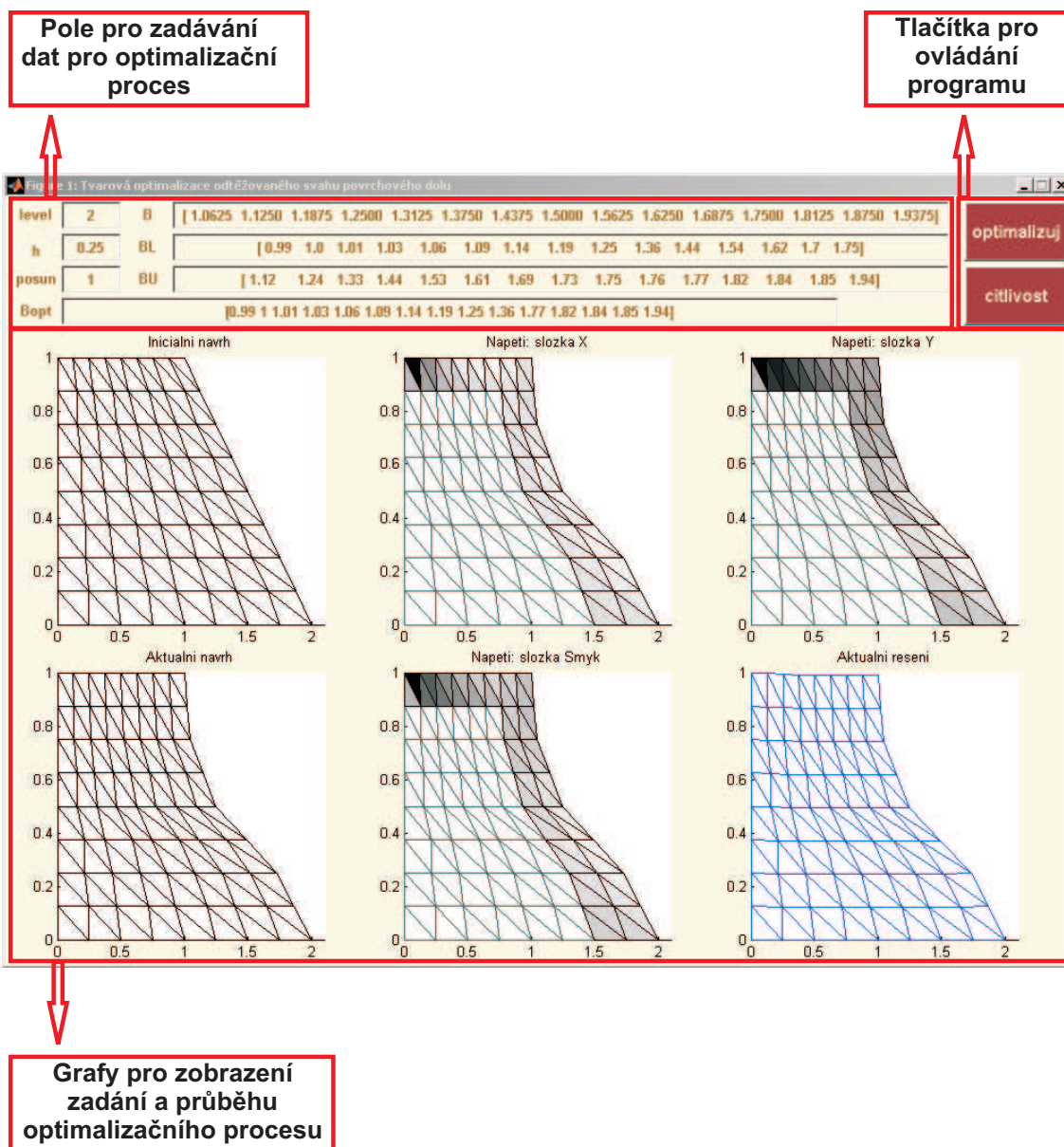
Známe-li řešení dopředu nebo z předchozích výpočtů, můžeme do pole  $B_{opt}$  přímo vepsat tuto hodnotu a provést pouze analýzu citlivosti tlačítkem *citlivost*.



Obrázek 30: Analýza citlivosti pro každou proměnnou z příkladu 2



Obrázek 31: Fragment svahu- Lom Bílina



Obrázek 32: GUI

## 9 Závěr

V této práci bylo ukázáno využití multigridní metody v úloze lineární pružnosti, přičemž bylo využito některých pěkných vlastností diskretizované oblasti a také bylo podstatně zefektivněno vyskládání matice tuhosti. Formulovali jsme poté úlohu tvarové optimalizace, v níž jsme aplikovali bundle trust metodu pro nalezení optimálních návrhových proměnných. Pro vypočtení subgradientu jsme použili numerického výpočtu, avšak je možné, že by bylo vhodnější využít analytického nebo alespoň semianalytického výpočtu, abychom měli k dispozici kvalitnější informaci o subgradientu a poté tyto přístupy mezi sebou porovnat. Po výpočtu optimalizační úlohy jsme přistoupili k analýze citlivosti na změnu jednotlivých proměnných cenového funkcionálu a ta nám přinesla ilustraci, že řešení, které jsme našli je v lokálním smyslu „optimální“. Tato analýza nám také potvrdila, že cenový funkcionál je nehladký a dokonce nalezená minima jsou v nehladkých bodech. Jako další možné pokračování v těchto typech úloh se nabízí například využití lepšího modelu zohledňující i plasticitu anebo namísto Dirichletových okrajových podmínek na hranicích aplikovat tření.

## 10 Reference

- [1] M.Brdička, L.Samek, B.Sopko, *Mechanika kontinua* (Academia, 2005).
- [2] O.C.Zienkiewicz, *The Finite Element Method in Engineering Science* (McGraw-Hill, London, 1971).
- [3] P.Beremlijski, *Řešení kontaktních úloh se třením pomocí bundle metod* (diplomová práce, 1999).
- [4] J.Drla, *Metody modelování křivek a ploch v počítačové geometrii* (Univerzita Palackého, Přírodovědecká fakulta, Olomouc, 1992).
- [5] P.Beremlijski, *Využití metod nehladké optimalizace v tvarové optimalizaci* (disertační práce, 2007).
- [6] J.Outrata, M.Kočvara, J.Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints* (Kluwer Academic Publishers, 1998).
- [7] J.Outrata, H.Schramm, J.Zowe, *Bundle Trust Methods: Fortran Codes for Nondifferentiable Optimization- User's Guide* (Report No. 269, Universität Bayreuth, 1991).
- [8] J.Zowe, *Nondifferentiable optimization* (University of Bayreuth, 1985).